agriculture

MDPI

*Article*

# MobileDBH: Estimating Tree Diameter at Breast Height from Smartphone Images Using a Lightweight Diffusion Depth Network for Field Tree Phenotyping

**Chao Mou [1,2,3], Jiahua Fan [1,2,3], Ang Liu [1,2,3], Chang Liu [1,2,3], Kecheng Huang [1,2,3], Huiyu Ding [1,2,3], Jingchen Li [1,2,3] and Haiyan Zhang [1,2,3,*]**

1 School of Information Science and Technology (School of Artificial Intelligence), Beijing Forestry University, Beijing 100083, China; chao_m@bjfu.edu.cn (C.M.); fjiahua2783@bjfu.edu.cn (J.F.); liuang@bjfu.edu.cn (A.L.); liuchanggg@bjfu.edu.cn (C.L.); huangkecheng@bjfu.edu.cn (K.H.); dinghuiyu@bjfu.edu.cn (H.D.); ljc20051113@bjfu.edu.cn (J.L.)
2 Engineering Research Center for Forestry-Oriented Intelligent Information Processing of National Forestry and Grassland Administration, Beijing 100083, China
3 Hebei Key Laboratory of Smart National Park, Beijing 100083, China
* Correspondence: zhyzml@bjfu.edu.cn

**Abstract**

Diameter at breast height (DBH) is a crucial indicator for obtaining tree phenotypes in orchard management, plantation monitoring, and agroforestry systems. LiDAR technology has high measurement accuracy, but it is costly and difficult to deploy flexibly in outdoor scenarios, while smartphones have emerged as a viable alternative due to their portability and low cost. In this paper, we propose a DBH estimation method based on monocular depth estimation, supported by a mobile application for algorithm deployment and result visualization. To address the limited computing resources on mobile devices, we design HR-DiffusionDepth, a lightweight diffusion-based monocular depth estimation network for smartphones, which generates pixel-wise 3D coordinates from a single image using camera intrinsics, thereby replacing LiDAR for DBH calculation. Experiments on the KITTI and SPREAD datasets show that HR-DiffusionDepth achieves the best depth estimation accuracy among similar lightweight models, reducing Abs Rel by up to 25.3% relative to the state-of-the-art (SoTA) lightweight baseline, with only 6.26 M parameters. The validation results show that the root mean square error (RMSE) of DBH estimation is 3.10 cm and the mean absolute error (MAE) is 2.25 cm, demonstrating the potential of this approach for agricultural scenarios such as orchards and plantations.

**Keywords:** tree phenotyping; lightweight monocular depth estimation; smartphone; DBH estimation; smart agroforestry



Academic Editor: Luca Ghiani

Received: 1 June 2026
Revised: 28 July 2026
Accepted: 29 July 2026
Published: 31 July 2026



## 1. Introduction

Diameter at breast height (DBH) is an important parameter for tree phenotyping and agricultural and forestry resource surveys. It provides key data for assessing tree growth, estimating biomass, and monitoring tree health status [1]. Traditional contact measurement methods, such as tape measures and calipers, are highly accurate but involve high labor costs and low efficiency and are easily constrained by forest terrain, making large-scale measurements inefficient. LiDAR, capable of capturing 3D point cloud data to precisely extract tree structural parameters, has been widely utilized for large-scale measurement of tree DBH [2]. However, conducting DBH measurements with LiDAR in field conditions

remains challenging. On one hand, transporting, deploying, and maneuvering LiDAR equipment with substantial size and weight in wild environments is difficult [3]. On the other hand, algorithms for processing the scanned point cloud data face difficulties such as challenging environmental filtering and heavy reliance on empirical methods for DBH estimation. As a result, developing a portable, low-cost, and user-friendly method for estimating tree diameter in the field is of great significance for tree phenotyping and smart agroforestry management.

Compared to bulky LiDAR scanning equipment, rapidly proliferating lightweight smartphones offer a low-cost and convenient solution for field DBH estimation. Smartphones have significant advantages in terms of data acquisition time, hardware cost, and ease of use, with data collection efficiency reaching mere seconds and device costs being approximately 1–2% of those of a LiDAR system [3,4]. However, due to hardware limitations, smartphones typically capture images without depth information. Since three-dimensional information is crucial for accurate DBH estimation, the precision of directly extracting DBH information from images often does not meet the requirements [5]. Therefore, extracting depth information from images to accurately estimate DBH is key to utilizing smartphones for field tree structural parameter acquisition.

Deep learning-based depth estimation algorithms can infer scene depth information from images by learning visual cues such as geometric constraints, spatial occlusion information, and changes in texture features. This provides an important technical foundation for employing smartphones in automatic DBH measurement. Typically, depth estimation includes two main types: multi-view depth estimation and monocular depth estimation [6]. Multi-view depth estimation infers depth information by matching images from different viewpoints. However, in tree scenes with similar texture features, using multiple overlapping images can make feature extraction and matching particularly challenging. Additionally, multi-view image matching requires high computational power [7], making it difficult to implement multi-view depth estimation on smartphones. For example, Fu et al. [7] proposed a binocular vision-based DBH estimation method that achieved high-precision measurements, but the method has high hardware requirements and cannot be ported to smartphones. Fortunately, monocular depth estimation algorithms do not require image alignment and calibration, which allows depth estimation with lower computational and energy demands. Thus, in this work, we deployed a DBH estimation model based on monocular depth estimation on smartphones.

With the rapid advancement of deep learning, researchers have introduced numerous neural network models to tackle monocular depth estimation tasks. Generally, depth estimation can be viewed as a pixel-wise regression problem. For instance, studies [8] have employed convolutional neural networks (CNNs) for depth estimation. Since CNNs are limited in capturing global information, researchers have used the self-attention mechanism in transformer architectures to leverage long-range dependencies in images, thereby improving the accuracy of depth estimation [9,10]. However, these regression models often suffer from severe overfitting and tend to lose detailed representations that are useful for depth estimation [11,12]. In recent years, diffusion models have been introduced into depth estimation tasks. The iterative process of diffusion models can effectively utilize global information and local details, producing smoother depth maps with better detail preservation [11,13,14]. The learning mechanism of diffusion models gradually recovers real data from noise, which enables the handling of more complex problems and reduces the tendency toward overfitting. Unfortunately, this learning mechanism also results in substantial computational costs when applying diffusion models to monocular depth estimation, making it impractical to deploy on resource-constrained mobile edge devices such as smartphones. Therefore, we propose a lightweight diffusion model for depth estimation

that maintains high estimation accuracy while significantly reducing parameter size and computational overhead, thereby enabling practical deployment on smartphones.

Additionally, during the sampling process, the encoder of diffusion models may inevitably lose edge information [11], while precise object boundary information can enhance depth estimation accuracy [12]. Therefore, we introduce an edge detail extraction branch specifically designed to capture edge details, enabling the diffusion model to focus on the boundary regions of objects and better preserve edge information during feature extraction. In summary, this paper presents a lightweight diffusion model that can be deployed on smartphones for low-cost, accurate DBH estimation in the field. Specifically, the contributions of this paper are as follows:

(1) A smartphone-based application for measuring tree DBH, named MobileDBH, was developed. Depth estimation is executed directly on the smartphone, while trunk segmentation currently relies on a cloud-based API. This application can complete the task of obtaining tree phenotypic parameters in agricultural and forestry scenarios efficiently and at low cost.
(2) We propose a novel lightweight monocular depth estimation network. By integrating the HF-HRNet module, which offers higher computational efficiency and fewer parameters, we significantly reduce the parameter size and computational overhead of the diffusion model, enabling it to operate efficiently in the limited computing resource environment of smartphones.
(3) We designed lightweight edge feature extraction and multi-scale feature extraction modules. By utilizing edge and multi-scale features to guide the depth estimation task, the approach captures critical edge information, maintaining high prediction accuracy while significantly reducing model parameter size and computational expense.

## 2. Related Work

### *2.1. DBH Estimation*

The development of DBH measurement technology can be roughly divided into three stages: traditional contact measurement, LiDAR-based measurement, and image-based deep learning estimation. Traditional contact measurement mainly relies on tools such as tape measures, diameter rulers, and calipers [15], which can provide precise single-tree measurement results and are widely used as reference standards for ground-truth values. However, it is inefficient, labor-intensive, and difficult to implement in complex terrain or dense forest stands, failing to meet the requirements of large-scale forest monitoring.

LiDAR-based measurement, especially terrestrial laser scanning (TLS), acquires high-precision three-dimensional point cloud data of trees by emitting and receiving laser pulses, significantly improving the efficiency of forest mapping [16]. Researchers can automatically or semi-automatically extract various tree parameters from the point cloud, including DBH. Liang et al. [17] demonstrated that the root mean square error of TLS automatic DBH measurement can be controlled within 2 cm, with high accuracy and reliability. However, TLS equipment is costly and bulky, and the data collection and point cloud processing procedures in the field are complex, restricting its application in routine forestry surveys [16].

Image-based measurement has received attention in recent years due to its low cost and high efficiency. With the popularity of smartphones and the advancement of deep learning technology, estimating DBH from a single image has become a research hotspot [18].

However, most existing image methods use deep learning for tree trunk segmentation rather than depth estimation, still relying on dedicated hardware to obtain depth information, and there is a lack of research on DBH measurement using monocular depth

estimation [19]. The main contribution of this paper lies in proposing a DBH measurement method based on depth estimation rather than tree trunk segmentation.

Recovering three-dimensional depth information from a single two-dimensional image is a recognized challenge, directly restricting the accuracy of DBH estimation. Diffusion models have made significant progress in the field of image generation and have demonstrated the ability to generate high-quality depth maps in tasks such as monocular depth estimation. However, existing diffusion models have high computational costs and are difficult to deploy on mobile devices such as smartphones. This paper introduces the lightweight design of diffusion models and applies it to field DBH estimation, significantly reducing the parameter count and computational cost while maintaining high accuracy. The feasibility of the scheme is verified through the designed lightweight network architecture, providing a new technical path for low-cost and portable smart agricultural and forestry tree diameter measurement.

Multiscale feature learning, attention-based recognition methods, and lightweight visual architectures have been studied in forestry, agriculture, and related visual tasks. Multiscale convolution and multi-layer feature fusion have been applied to remote sensing forest species recognition [20], and a method combining dilated pyramid attention and spatial enhancement has also been proposed for the same task [21]. In the field of crop phenotyping, the Tswin-F network integrates bilateral local attention and self-supervised learning to achieve multi-scale feature aggregation in tomato leaf disease recognition [22]. These three works indicate that multi-scale feature extraction, attention enhancement, and hierarchical feature fusion have reference value for fine-grained visual tasks such as tree DBH estimation. At the same time, HGCS-Det combines attention calibration and boundary enhancement for object detection in complex scenes [23], and GMamba is a lightweight state-space architecture for image classification [24]; these lightweight architectures demonstrate the application potential of lightweight design in complex scene recognition and deployment on resource-constrained devices, although HGCS-Det and GMamba are not specifically developed for agricultural and forestry applications.

### *2.2. Edge-Based Depth Estimation*

The goal of monocular depth estimation is to infer scene depth from a single RGB image, which is an important research topic in computer vision. In recent years, deep learning methods have made significant progress on this task [12]; however, the generated depth maps still tend to exhibit blurriness, excessive smoothing, or large local deviations near object boundaries, leading to loss of fine details. For applications such as DBH estimation that rely on accurate object contours, the quality of edge information directly affects subsequent measurement results.

To address the aforementioned issue, existing research has attempted to explicitly incorporate image edge information into depth estimation networks. These methods typically rely on the prior assumption that high-frequency edges in an image are strongly associated with discontinuities in scene depth—namely, object boundaries. Proper utilization of this prior can help improve the prediction accuracy of depth maps in boundary regions and enhance edge sharpness.

Based on this approach, researchers have proposed various implementation methods. Among them, multi-task learning is a relatively common method. Its basic idea is to enable the network to simultaneously predict depth maps and edge maps and to strengthen the constraints between the two tasks by sharing feature representations or joint loss functions, thereby improving the depth prediction effect at the object boundaries. Some studies have also set up independent edge extraction branches, integrating the extracted edge features with the depth features in the main network to guide the generation of the depth map.

For example, Xu et al. [25] introduced the edge map as prior information into the structured attention-guided conditional random field framework, and improved the sharpness of the depth map boundaries through global optimization; Ramamonjisoa and Lepetit [26] proposed a differentiable variational refinement framework, using the predicted edge map to impose regularization constraints on the energy function, making the depth gradient consistent with the image edge as much as possible, thereby refining the initial depth map.

Based on the above research, this paper introduces the edge-guiding mechanism into the DBH estimation task. In this task, the clarity of the trunk contour directly affects the measurement accuracy. Therefore, HR-DiffusionDepth is designed with a lightweight edge feature extraction module, which incorporates low-level geometric edge information into the denoising process of the diffusion model. While controlling the computational cost, it generates a depth map with clearer trunk boundaries and higher depth accuracy.

### *2.3. Lightweight Depth Estimation Models*

The research goal of lightweight depth estimation is to maintain high prediction accuracy in scenarios with limited computing resources, such as on mobile devices. The related methods are mainly divided into two learning paradigms: self-supervised and supervised. These methods usually reduce computational costs by simplifying network structures, optimizing feature extraction, or integrating lightweight modules to meet the requirements of efficient deployment.

In the self-supervised learning direction, researchers train on unlabeled data to reduce reliance on large-scale labeled datasets and optimize network structures to reduce computational costs. Godard et al. [27] proposed Monodepth2, which introduced three strategies: minimum reprojection loss, full-resolution multi-scale sampling, and automatic masking loss. This model is representative in this direction. Based on this, Zhang et al. [28] proposed Lite-Mono, which combines the characteristics of CNN and Transformer to extract multi-scale local features through consecutive expansion convolution modules and encode long-range dependencies through a local-global feature interaction module. This model surpasses Monodepth2 in terms of accuracy while reducing the parameter count by approximately 80%, making it more suitable for mobile device deployment.

In the supervised learning direction, researchers mainly reduce model complexity through architecture design and lightweight strategies. Wofk et al. [29] proposed FastDepth, which adopts an efficient encoder-decoder structure and combines network pruning to reduce inference latency. Li et al. [30] proposed MobileDepth, which extracts local features using MobileNetV2, captures global context through the DSAB module, and limits the computation of self-attention, significantly compressing the parameter size compared to models based on ViT, and is suitable for deployment on edge devices. Hu et al. [31] used the knowledge distillation method, leveraging auxiliary unlabeled data to guide the student model to learn from the teacher model, achieving performance similar to the current best methods with only about 1% of the parameter size.

Existing lightweight depth estimation models have significantly improved in terms of parameter size and computational efficiency but still have deficiencies in global context modeling and edge detail preservation, making it difficult to meet the requirements of DBH estimation for trunk boundary accuracy. To address these issues, this paper simplifies the architecture and improves the modules based on the diffusion model for estimation. It proposes HR-DiffusionDepth, which integrates a hardware-friendly HF-HRNet module and a dedicated edge feature extraction branch, and utilizes the global modeling ability of the diffusion process to reduce model complexity while maintaining high depth estimation accuracy.

### *2.4. Dense Visual Prediction Diffusion Model*

The Dense Visual Prediction Diffusion Model (DDP) [32] is a dense visual prediction framework built upon a conditional diffusion pipeline. Its central principle is to leverage the progressive denoising mechanism of diffusion models, breaking down the complex mapping task into a series of manageable, iterative steps. In contrast to conventional regression-based approaches, this "noise-to-map" generative paradigm concurrently leverages global structural information and local fine-grained features during prediction. This enables the generation of outputs that are both globally coherent and rich in local detail. For depth estimation, DDP employs a high-capacity encoder to extract multi-scale spatial features [33], which are then guided by conditional information throughout the iterative denoising process to recover a more precise depth distribution. The efficacy of this mechanism has been proven in various dense prediction tasks, including semantic segmentation and bird's-eye-view (BEV) map generation, indicating its significant potential for depth estimation.

Nevertheless, the direct application of DDP to monocular depth estimation, particularly on resource-constrained edge devices such as smartphones, presents formidable challenges. Firstly, the original DDP architecture typically depends on a Swin Transformer encoder [33] and a standard feature pyramid network (FPN) [34]. While this combination excels at capturing rich spatial and contextual information, it incurs substantial computational overhead and a large parameter footprint, resulting in high inference latency that renders it impractical for mobile deployment. Secondly, the encoder within the diffusion model is prone to losing critical edge information during successive downsampling operations. Given that object boundaries are of paramount importance in depth estimation, the blurring of trunk contours in DBH estimation tasks will directly degrade measurement accuracy [35]. Consequently, without targeted architectural optimizations, the utility of DDP for efficient measurement in complex environments, such as forested areas, is severely constrained.

To address the aforementioned issues, this study proposes a lightweight transformation and edge feature enhancement solution within the DDP framework, namely the HR-DiffusionDepth model. On one hand, we introduce HF-HRNet [36], which boasts higher computational efficiency and is more compatible with mobile Neural Processing Units (NPUs), to replace the Swin Transformer as the multi-scale feature extractor, thereby significantly reducing the model's parameter count and computational overhead. On the other hand, an edge feature branch is added during the feature extraction stage to effectively alleviate edge blurring issues. Through these design innovations, we not only retain the strengths of DDP in global and local information modeling but also overcome its shortcomings in mobile deployment and boundary detail recovery, ultimately achieving low-cost, single-shot, and high-precision DBH estimation in field scenarios.

## 3. Materials and Methods

### *3.1. The Framework of MobileDBH*

In this study, we propose MobileDBH, a low-cost mobile method for tree DBH estimation based on monocular depth estimation. MobileDBH utilizes a smartphone for RGB image acquisition and then leverages a deep learning model deployed on the device to perform depth estimation, parameter calibration, and DBH estimation, as shown in Figure 1.

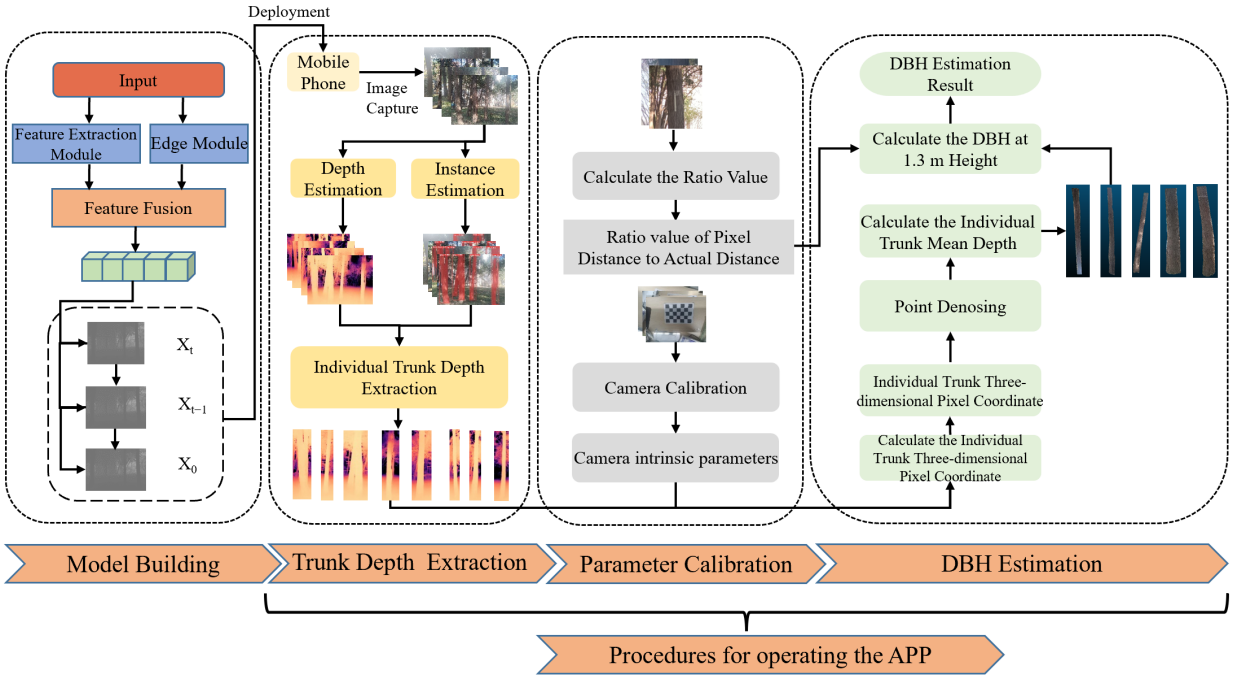


**Figure 1.** The framework of MobileDBH. The workflow consists of model building, trunk depth extraction, parameter calibration, and DBH estimation.

As illustrated in Figure 1, the specific workflow of MobileDBH comprises the following steps:

(a) Model Building. To estimate the 3D information of forest environments from visible light images for high-precision DBH estimation, we propose a lightweight depth estimation model named HR-DiffusionDepth based on the diffusion mechanism. Through a lightweight design, HR-DiffusionDepth, which leverages high-precision diffusion models, can be deployed on mobile devices. Furthermore, by incorporating multi-scale and edge feature extraction modules to realistically reconstruct the 3D environment of actual scenes and obtain sharper tree edges, the smartphone equipped with HR-DiffusionDepth can effectively overcome the challenge of imprecise DBH estimation caused by blurry tree edges.

(b) Trunk Depth Extraction. First, the Segment Anything Model (SAM, ViT-L variant, approximately 308 M parameters) [37], an off-the-shelf zero-shot instance segmentation model accessed via its official API, is utilized to obtain trunk mask information from images captured by the smartphone, enabling precise identification of trunk contours within the image. Simultaneously, the HR-DiffusionDepth model estimates the depth information of the entire scene. Subsequently, the individual trunk depth information is obtained by combining the acquired trunk mask with the scene depth information.

(c) Parameter Calibration. The intrinsic camera parameters of the smartphone are obtained using Zhang's calibration method [38], with an A4-printed checkerboard pattern ($9 \times 6$ internal corners, approximately 25 mm per square) used as the calibration target. Since the depth values output by HR-DiffusionDepth are of relative scale and do not directly correspond to real physical units, in this paper, a reference object (e.g., a ruler) is attached to the tree trunk, and the camera intrinsics are used to reconstruct a three-dimensional point cloud from the relative depth map, in which the two endpoints of the reference object are identified. The three-dimensional distance between the two points is computed, and the scale factor is obtained by dividing the known actual length of the reference object by this distance.

Specifically, let the image coordinates of a given pixel be $(u, v)$, with the corresponding relative depth value $d$. Let the focal lengths in the camera intrinsics be $f_x$ and $f_y$, and the principal point coordinates be $(c_x, c_y)$. The three-dimensional coordinate obtained by back-projecting this pixel is then

$$X = \frac{(u - c_x)d}{f_x}, \quad Y = \frac{(v - c_y)d}{f_y}, \quad Z = d. \tag{1}$$

Let the endpoints of the reference object correspond to pixel coordinates $(u_1, v_1, d_1)$ and $(u_2, v_2, d_2)$, and let the three-dimensional coordinates $P_1$ and $P_2$ be obtained by back-projection according to the above formula; the three-dimensional distance between the two points is $\|P_1 - P_2\|_2$. Since the pixel offsets in the back-projection formula are each scaled according to their respective depth value $d$, rather than by a uniform pixel ratio, points farther from or closer to the camera are each corrected according to their true depth when converted to three-dimensional coordinates. Thus, $\|P_1 - P_2\|_2$ reflects a relative three-dimensional scale that is unaffected by the distance between the object and the camera. The scale factor is obtained by dividing the known actual physical length $L$ of the reference object by this distance:

$$s = \frac{L}{\|P_1 - P_2\|_2}. \tag{2}$$

This scale factor is used only to correct the overall scale ambiguity of depth, i.e., the proportional relationship between relative and absolute depth; it does not involve perspective correction at the pixel-coordinate level, as this has already been accounted for in the back-projection formula. Therefore, $s$ applies uniformly across the entire scene and does not introduce deviations arising from differences in distance between trees and the camera. In this study, a scene is defined at the operational level as a range where the camera is fixed at the same position and multiple photos are taken by simply re-adjusting the framing (such as rotating the lens) without any physical movement. In field measurements, the effective shooting distance between the camera and the tree trunk was approximately 1 to 3 m. Measurements of multiple trees within the same scene do not require recalibration; when the camera is moved to a new position, it constitutes a new scene, and a new ruler calibration is necessary.

(d) DBH Estimation. By combining the depth information of the tree trunks with the camera intrinsics, we generate three-dimensional coordinates for each tree; the three-dimensional coordinates corresponding to the 1.3-m breast-height position are calculated from the depth information and the known ground-reference height, and are then back-projected onto the two-dimensional image. On this basis, outliers in the trunk point cloud ($|\text{depth} - \mu| > 2\sigma$) are removed using the Z-score method to denoise the point cloud. The average depth $\bar{d}$ of the denoised point cloud is used as the representative depth. The left and right boundary pixel points of the trunk are back-projected into three-dimensional coordinates using $\bar{d}$ and the camera intrinsics, and the Euclidean distance between the two gives the trunk width $w$. Combined with the scale factor, the DBH is calculated as follows:

$$\text{DBH} = w \times s. \tag{3}$$

### *3.2. HR-DiffusionDepth*

Inspired by DDP, we developed a lightweight depth estimation framework named HR-DiffusionDepth based on the diffusion mechanism. HR-DiffusionDepth enables MobileDBH, when deployed on a smartphone, to perceive 3D environmental information for the accurate estimation of tree DBH. As illustrated in Figure 2, the architecture of HR-

DiffusionDepth is composed of three main components: feature extraction module , feature fusion module, and denoising module. In HR-DiffusionDepth, the feature extraction module extracts multi-scale and edge features, which not only contain the global information of the image but also encompass rich local details. The feature fusion module then effectively fuses these multi-scale and edge features to generate richer and more representative feature representations. Finally, the fused features are fed into the denoising module as visual conditioning. Its core function is to provide a crucial constraint and guidance for the iterative denoising process of the conditional diffusion model. Specifically, it ensures that the output generated at each denoising step strictly adheres to the given visual information, thereby guaranteeing the accurate and stable reconstruction of a high-fidelity depth scene.

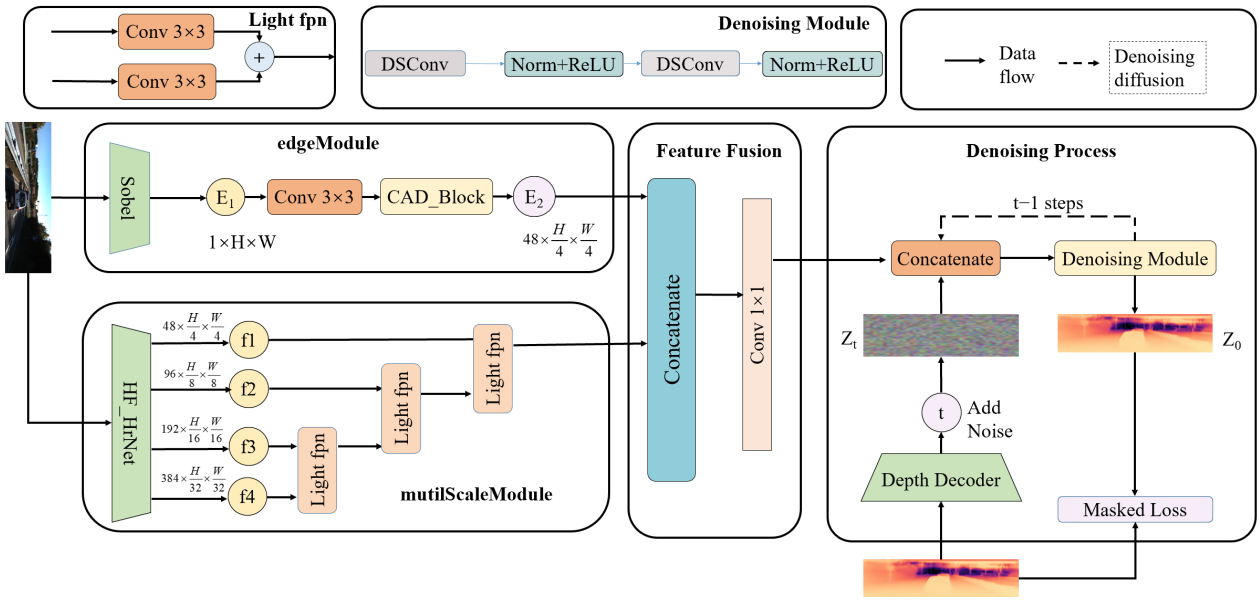


**Figure 2.** The architecture of HR-DiffusionDepth.

### 3.2.1. Feature Extraction Module

To enhance the perception of 3D environmental information and capture sharp tree edge details, we designed a feature extraction module capable of extracting multi-scale and edge features effectively while maintaining a low model parameter count. This module comprises two parallel feature extraction branches: (a) a multi-scale feature extraction branch called MultiScaleModule and (b) an edge feature extraction branch called EdgeModule. While receiving the input image $x \in \mathbb{R}^{H\times W\times 3}$, the feature extraction module simultaneously outputs a spatial feature map $F$ and an edge feature map $E$.

In the MultiScaleModule branch, a multi-scale feature encoder, such as HF-HRNet, outputs multi-scale features at four different resolutions, which are then fused using a lightweight feature pyramid network called lightFPN, resulting in a spatial feature map $F$ of size $H/4 \times W/4 \times 256$. The EdgeModule branch outputs an edge feature map $E$ of size $H/4 \times W/4 \times 48$. The computational processes of these two branches are elaborated in the following sections.

#### a. The MultiScaleModule Branch

The proposed MultiScaleModule branch consists of two stages: multi-scale feature extraction (Stage 1) and multi-scale feature fusion (Stage 2). Generally, diffusion-based depth estimation networks (e.g., DDP) predominantly employ a Swin Transformer encoder [33] and a standard FPN to learn the spatial structural information of images and perform feature fusion. This design incurs significant computational costs for both training and

inference, making it challenging to deploy on resource-constrained smartphone devices. Therefore, we developed lightweight optimizations for these two modules, aiming to reduce model parameters and computational complexity while preserving feature extraction performance as much as possible. In Stage 1, we replaced the Swin Transformer encoder with the more computationally efficient HF-HRNet. As a multi-scale feature extractor, HF-HRNet offers the following advantages: First, it achieves excellent feature extraction performance with relatively small parameter count and computational complexity. Second, it provides robust support for NPU acceleration functions in mobile devices, significantly enhancing operational efficiency on edge mobile devices. In summary, HF-HRNet not only possesses outstanding multi-scale feature extraction capabilities but also effectively optimizes model deployment performance on hardware through the introduction of a Hardware-aware Unit Module (HUM), thereby substantially boosting inference speed. As illustrated in Figure 2, HF-HRNet performs feature extraction on an input image of size $H \times W \times 3$, ultimately outputting four feature maps of different resolutions, denoted as $F_1$, $F_2$, $F_3$, and $F_4$ as shown in Equation (4).

$$F_i \in \mathbb{R}^{\left(H/2^{i+1}\right)\times\left(W/2^{i+1}\right)\times\left(48\times 2^{i-1}\right)} \tag{4}$$

where $i = 1, 2, 3, 4$ denote feature maps at different scales.

In Stage 2, we designed a lightweight module called lightFPN to facilitate multi-scale feature fusion. By replacing the $3 \times 3$ standard convolutional smoothing layer in the standard FPN module with a simple and lightweight $3 \times 3$ depth-wise separable convolution, we significantly reduced the parameter scale and computational cost of the module. As illustrated in Equation (5), lightFPN processes feature maps of different scales individually through each layer's $3 \times 3$ depth-wise separable convolution. Subsequently, the lower-resolution feature maps are upsampled by a factor of 2 and added to their corresponding higher-resolution counterparts. Finally, a smoothing convolutional layer is applied to eliminate artifacts and discontinuities for achieving progressive feature fusion.

$$X = \mathrm{DSConv}\left(\mathrm{Conv}(F_{i-1}) + \mathrm{Up}(\mathrm{Conv}(F_i))\right) \tag{5}$$

where DSConv$(\cdot)$ denotes the $3 \times 3$ depth-wise separable convolution operation, Conv$(\cdot)$ denotes the $1 \times 1$ standard convolution operation, and Up$(\cdot)$ denotes the $2\times$ upsampling operation.

b. The EdgeModule Branch

To enhance the model's edge perception capability, the EdgeModule branch integrates the Sobel edge detection algorithm [35] with the computationally lightweight HF-HRNet to efficiently extract fine-grained edge information. Since the Sobel edge detection algorithm focuses on high-frequency information such as contours and texture variations within the image, it effectively filters redundant color and illumination details. This enables the model to more readily capture critical features like object shapes and boundaries. The Sobel edge detection algorithm employs two $3 \times 3$ convolution kernels (i.e., Sobel operators) to detect gradient changes in the horizontal and vertical directions, respectively, thereby identifying object edges. Specifically, to minimize computational load on mobile devices, the EdgeModule branch applies the Sobel edge detection algorithm to generate an edge feature vector $E_1 \in \mathbb{R}^{H\times W\times 1}$. Subsequently, the CAD_Block module from the efficient HF-HRNet is utilized to perform downsampling, yielding a feature map $E_2 \in \mathbb{R}^{48\times H/4\times W/4}$.

#### 3.2.2. Fusion Module

As illustrated in Equation (6), the Fusion Module functions to fuse the obtained multi-scale feature maps $F$ and the edge feature map $E$. While $F$ is rich in spatial and semantic information, its successive downsampling operations result in a loss of high-frequency details, such as sharp edges. Conversely, the edge feature map $E$ offers precise boundary information but is devoid of the broader spatial and semantic context. To capitalize on the complementary strengths of both, the Fusion Module merges them to generate a new feature map that encapsulates edge, spatial, and semantic information. This module is composed of a concatenation operation followed by a Conv block. As depicted in Figure 2, the Fusion Module outputs a final fused feature map $C$, which has 256 channels.

$$C = \text{Conv}\left(\text{concat}(F, E)\right) \tag{6}$$

where concat$(\cdot)$ denotes the concatenation operation, and Conv$(\cdot)$ denotes the $1 \times 1$ convolution operation.

#### 3.2.3. Denoising Module

The diffusion model formulates the depth estimation problem as an iterative denoising process. The objective of the denoising module is to learn how to recover clear and accurate depth information from the noisy depth maps. Since the feature map $C$, learned by the feature extraction module, encapsulates a series of complex features, i.e., fusing edge features, spatial features, and semantic features, it serves as rich prior knowledge. In each iteration, the denoising module leverages $C$ as a visual guidance condition within the conditional diffusion model to reconstruct the image's depth information. As formalized in Equation (7), the module accepts the noisy sample $y_t$ at the current timestep $t$ and the visual condition $C$ as its inputs. By concatenating $y_t$ with the feature map $C$, the module engages in the denoising process to predict the distribution of the sample $y_{t-1}$ from the preceding timestep $t-1$.

$$p_\theta(y_{t-1} \mid y_t, C) \tag{7}$$

where $C$ represents the fused feature map, which serves as the visual condition for the diffusion model, and $\theta$ denotes the denoising module, i.e., the entire denoising network.

Notably, to reduce the number of parameters and computational overhead, we replace the six-layer deformable attention block in the DDP with a lightweight stack of two depth-wise separable convolutional layers. Each of these convolutional layers consists of a $3 \times 3$ depth-wise separable convolution, a batch normalization (BatchNorm) layer, and a ReLU activation.

#### 3.2.4. Training Process

As illustrated in Algorithm 1, given an original image $I$ and its corresponding ground-truth depth map $M$, the original image $I$ is first processed by the EdgeModule and the MultiScaleModule to obtain an edge map $E$ and a multi-scale feature map $F$, respectively. These features are then fused through the feature fusion module to generate a conditional feature map $C$, which serves as the latent space representation.

**Algorithm 1** Denoising diffusion training process

**Input:** Input images $I$, ground-truth depth maps $M$
**Output:** Training loss $\mathcal{L}$
$F \leftarrow \text{MultiScaleExtraction}(I)$ {Extract multi-scale features}
$E \leftarrow \text{EdgeExtraction}(I)$ {Extract edge features}
$C \leftarrow \text{Concatenate}(F, E)$
$y \leftarrow \text{Encoding}(M)$ {Encode ground truth}
$t \sim \mathcal{U}(0,1), \epsilon \sim \mathcal{N}(0,1)$ {Sample timestep and noise}
$y_t \leftarrow \sqrt{\alpha_{\text{cumprod}}(t)} \cdot y + \sqrt{1-\alpha_{\text{cumprod}}(t)} \cdot \epsilon$
$y_{\text{pred}} \leftarrow \text{DenoisingModule}(y_t, C, t)$
$\mathcal{L} \leftarrow \text{ObjectiveFunction}(y_{\text{pred}}, M)$
**return** $\mathcal{L}$

Concurrently, the ground-truth depth map $M$ is progressively corrupted by adding Gaussian noise, generating a sequence of noisy depth map samples, $y_t$, at varying noise levels. Finally, the denoising module takes both the latent feature map $C$ and the noisy depth map $y_t$ as input. Through an iterative, step-by-step denoising process, it recovers the original depth information, yielding the predicted depth map $y_{\text{pred}}$. During the inference stage, this process is initiated with pure Gaussian noise instead of a noisy version of the ground-truth depth map.

#### 3.2.5. Loss Function

The objective function in Algorithm 1 (corresponding to the masked loss labeled in Figure 2) is implemented using masked scale-invariant logarithmic loss, following the formula format in AdaBins. Given the predicted depth $\hat{d}$ and the true depth $d$, the loss is calculated only on the valid pixels, which are defined as pixels satisfying $0 < d \leq d_{\max}$, and invalid pixels (such as missing LiDAR returns) are excluded through the mask.

Let $g = \log(\hat{d} + \epsilon) - \log(d + \epsilon)$, where $\epsilon = 0.001$ is used to avoid numerical instability when the input is close to zero. The loss function is defined as

$$\mathcal{L} = \sqrt{\text{Var}(g) + 0.15 \cdot (\overline{g})^2} \tag{8}$$

where $\text{Var}(g)$ and $\overline{g}$ represent the variance and mean of $g$ on the valid pixels, respectively.

The scaling factor $\gamma$ used in the normalization step of Algorithm 1 corresponds to the bit-scale hyperparameter in the implementation, and it is set to 0.1 in this paper's implementation. The scaling constant $\alpha$ (typically set to 10 in some implementations of this loss) is not included in the implementation presented in this paper.

#### 3.2.6. Diffusion Process: Encoding, Noise Scheduling, and Sampling

Let the ground-truth depth map be denoted as $M$. The encoding operation is defined as

$$\text{Encoding}(M) = \left( \frac{M - d_{\min}}{d_{\max} - d_{\min}} \times 2 - 1 \right) \times \gamma \tag{9}$$

where $d_{\min}, d_{\max}$ denote the known depth range of the dataset (e.g., 0–80 m for KITTI), and $\gamma$ is the scaling factor defined in Section 3.2.5.

The noise scheduling strategy adopts a cosine schedule, with $\alpha_{\text{cumprod}}(t)$ defined as

$$\alpha_{\text{cumprod}}(t) = \cos^2\left( \frac{t + n_s}{1 + d_s} \times \frac{\pi}{2} \right) \tag{10}$$

where $n_s = 0.0002$ and $d_s = 0.00025$ are small constants controlling the smooth transition at both ends of the scheduling curve. The physical meaning of $\alpha_{\text{cumprod}}(t)$ is the proportion

of the true signal (the encoded depth $y$) retained in the noisy sample $y_t$ at timestep $t$, with a value range of $(0, 1)$: it approaches 1 as $t \to 0$ (nearly noise-free) and approaches 0 as $t \to 1$ (nearly pure noise).

During inference, $y_t$ is initialized from standard Gaussian noise $\epsilon \sim \mathcal{N}(0, 1)$. At each sampling step, the model takes the current noisy sample $y_t$ and the visual condition $C$ as input to predict the denoised estimate $y_{\text{pred}}$. The distribution $y_{t-1}$ is then computed according to the DDIM sampling formula, using the $\alpha_{\text{cumprod}}$ values at the current and next timesteps:

$$\epsilon_{\text{pred}} = \frac{y_t - \sqrt{\alpha_{\text{cumprod}}(t)} \cdot y_{\text{pred}}}{\sqrt{1 - \alpha_{\text{cumprod}}(t)}}, \quad y_{t-1} = \sqrt{\alpha_{\text{cumprod}}(t-1)} \cdot y_{\text{pred}} + \sqrt{1 - \alpha_{\text{cumprod}}(t-1)} \cdot \epsilon_{\text{pred}} \tag{11}$$

This process is iterated until $t = 0$, yielding the final depth prediction; in the default configuration of this paper, the number of sampling steps is 1.

### *3.3. Experimental Setup*

#### 3.3.1. Dataset

This paper comprehensively evaluated the depth estimation performance of HR-DiffusionDepth on two outdoor datasets, KITTI [39] and SPREAD [40]. The KITTI dataset is the mainstream benchmark in the field of stereo vision algorithm evaluation, containing a large number of stereo image pairs and corresponding LiDAR sparse depth ground truth collected by vehicle-mounted cameras. The image pairs record the real urban road scenes from slightly offset horizontal perspectives, providing important data support for the depth estimation task. This paper adopted the standard Eigen partition method [8], using 23,158 image pairs for training and 697 pairs for testing.

SPREAD is a large-scale, high-fidelity synthetic dataset specifically designed for forest-related machine learning tasks [40]. It includes RGB images and corresponding depth information of trees in 11 scenarios, with a total of 36,170 images. The RGB images have a resolution of 960 × 540, and the depth images have a resolution of 480 × 270. The dataset is split by scenario, with eight scenarios (34,370 images) used for training and the remaining three scenarios (1800 images) held out for testing. The model training stops after a fixed number of pre-defined iterations, and the obtained checkpoint is used directly. The hyperparameters follow the settings used in KITTI training.

In addition, this paper collected multiple sets of images on the campus of Beijing Forestry University in 2025. 20 representative trees were selected to construct a field scene test set for the DBH estimation case analysis. The selected DBH values ranged from 11.46 to 30.86 cm.

#### 3.3.2. Evaluation Indicators

As shown in Table 1, this paper used the standard evaluation metrics in the field of depth estimation to evaluate the depth estimation model, including absolute relative error (Abs Rel), squared relative error (Sq Rel), root mean square error (RMSE), log root mean square error (RMSE log), and threshold accuracy ($\delta$). Among them, Abs Rel, Sq Rel, RMSE, and RMSE log are error-related indicators, and the lower the value, the better; $\delta < 1.25$, $\delta < 1.25^2$, and $\delta < 1.25^3$ are accuracy-related indicators, and the higher the value, the better. These indicators comprehensively evaluate the degree of agreement between the predicted depth and the true depth from the perspectives of error and accuracy. In addition, as shown in Table 2, this paper used root mean square error (RMSE) and mean absolute error (MAE) to evaluate the accuracy of the DBH estimation method. On the KITTI and SPREAD datasets, the model was trained within the known depth range of each dataset,

using the actual metric depth as the direct supervision signal. Abs Rel and RMSE were computed by directly comparing the model's raw output with the ground-truth depth.

**Table 1.** Evaluation metrics for the accuracy of depth estimation.

| Metric | Formula |
| --- | --- |
| Abs Rel | $\frac{1}{N}\sum_{i=1}^{N}\frac{\lvert d_i-\hat{d}_i\rvert}{d_i}$ |
| Sq Rel | $\frac{1}{N}\sum_{i=1}^{N}\frac{(d_i-\hat{d}_i)^2}{d_i}$ |
| RMSE | $\sqrt{\frac{1}{N}\sum_{i=1}^{N}(d_i-\hat{d}_i)^2}$ |
| RMSE log | $\sqrt{\frac{1}{N}\sum_{i=1}^{N}(\log d_i-\log \hat{d}_i)^2}$ |
| $\delta <$ thr | % of $d$ satisfies $\max\left(\frac{\hat{d}_i}{d_i},\frac{d_i}{\hat{d}_i}\right) <$ thr, for thr $\in \{1.25, 1.25^2, 1.25^3\}$ |

Note: $d$ denotes the depth value of a pixel, $N$ represents the total number of pixels, and $i$ is the index of the pixel.

**Table 2.** Evaluation metrics for the accuracy of DBH estimation.

| Metric | Formula |
| --- | --- |
| RMSE | $\sqrt{\frac{1}{n}\sum_{i=1}^{n}(a_i-\hat{a}_i)^2}$ |
| MAE | $\frac{1}{n}\sum_{i=1}^{n}\lvert a_i-\hat{a}_i\rvert$ |

Note: $a$ denotes the DBH of a tree, $n$ represents the total number of trees, and $i$ is the index of the tree.

#### 3.3.3. Implementation Details

This paper implemented the HR-DiffusionDepth model based on the PyTorch 1.12.0 with CUDA 11.6 [41]. When training on the KITTI dataset, the batch size was set to 16, and 40 epochs were trained; when training on the SPREAD dataset, the batch size was set to 8, and 30 epochs were trained. Additionally, we set the depth range of KITTI and SPREAD to 0–80 m and 0–50 m, respectively. We used the AdamW optimizer to train HR-DiffusionDepth, with the momentum parameters $\beta_1$ and $\beta_2$ configured as 0.95 and 0.99, respectively, and the initial learning rate set to 0.0001. During the training process, we applied random cropping, random rotation, color jittering, and other operations to enable the model to learn more robust and generalizable features, thereby improving the performance of the model.

## 4. Results

### *4.1. Comparative Experiments*

#### 4.1.1. Comparison with Representative Lightweight Methods

a. Evaluation Results on the KITTI Dataset.

On the KITTI benchmark dataset, we systematically evaluated the performance of HR-DiffusionDepth under low-resolution (640 × 192) and high-resolution (1024 × 320 and 1280 × 384) input conditions, and conducted a comprehensive comparison with the state-of-the-art lightweight supervised and unsupervised methods. The results are summarized in Tables 3 and 4.

From Tables 3 and 4, it can be seen that regardless of the low-resolution or high-resolution input, HR-DiffusionDepth achieved the lowest Abs Rel and RMSE values, while maintaining the highest threshold accuracy ($\delta < 1.25$, $\delta < 1.25^2$, $\delta < 1.25^3$), confirming its strong depth estimation performance. Particularly, at the resolution setting of 1280 × 384, the Abs Rel and RMSE of HR-DiffusionDepth dropped to 0.068 and 2.730 respectively, and the corresponding $\delta < 1.25$, $\delta < 1.25^2$, $\delta < 1.25^3$ accuracies reached 0.944, 0.991, and 0.997 respectively, highlighting its accuracy in depth estimation. Compared with the

best-performing TinyDepth, HR-DiffusionDepth reduced Abs Rel and RMSE by 25.3% and 32.5% respectively, and the precision indicators $\delta < 1.25$, $\delta < 1.25^2$, $\delta < 1.25^3$ increased by 3.2%, 2.1%, and 1.2% respectively, demonstrating significant advantages.

**Table 3.** Low-resolution quantitative comparison on the KITTI dataset.

| Method | Type | Input Resolution | Abs Rel ↓ | Sq Rel ↓ | RMSE ↓ | RMSE log ↓ | $\delta < 1.25$ ↑ | $\delta < 1.25^2$ ↑ | $\delta < 1.25^3$ ↑ | Params (M) |
|---|---|---|---|---|---|---|---|---|---|---|
| 640 × 192 | | | | | | | | | | |
| DDVO [42] | M | 640 × 192 | 0.151 | 1.257 | 5.583 | 0.228 | 0.810 | 0.936 | 0.974 | 28.1 |
| GeoNet [43] | M | 640 × 192 | 0.149 | 1.060 | 5.567 | 0.226 | 0.796 | 0.935 | 0.975 | 31.6 |
| Monodepth2 [27] | M | 640 × 192 | 0.110 | 0.831 | 4.642 | 0.187 | 0.879 | 0.961 | 0.982 | 14.3 |
| HR-Depth [44] | M | 640 × 192 | 0.109 | 0.792 | 4.632 | 0.185 | 0.884 | 0.962 | 0.983 | 14.7 |
| CAD-Depth [45] | M | 640 × 192 | 0.110 | 0.812 | 4.686 | 0.187 | 0.882 | 0.962 | 0.983 | 18.8 |
| Lite-Mono [28] | M | 640 × 192 | 0.107 | 0.765 | 4.561 | 0.183 | 0.886 | 0.963 | 0.983 | 3.1 |
| DIFFNet [46] | M | 640 × 192 | 0.102 | 0.764 | 4.483 | 0.180 | 0.896 | 0.965 | 0.983 | 10.9 |
| MonoViT [9] | M | 640 × 192 | 0.099 | 0.708 | 4.372 | 0.175 | 0.900 | 0.967 | 0.984 | 27.0 |
| TinyDepth(S) [47] | M | 640 × 192 | 0.096 | 0.665 | 4.249 | 0.171 | 0.904 | 0.968 | 0.985 | 6.2 |
| RA-Depth [48] | M | 640 × 192 | 0.096 | 0.632 | 4.216 | 0.171 | 0.903 | 0.968 | 0.985 | 10.0 |
| TuMDE [49] | D | 640 × 192 | 0.148 | 0.890 | 5.282 | - | 0.813 | 0.954 | 0.987 | 3.4 |
| FastDepth [29] | D | 640 × 192 | 0.150 | 0.890 | 5.321 | - | 0.808 | 0.945 | 0.981 | 3.9 |
| GuidedDepth [50] | D | 640 × 192 | 0.119 | 0.771 | 4.456 | - | 0.857 | 0.965 | 0.990 | 5.8 |
| TST [51] | D | 640 × 192 | 0.114 | 0.649 | 4.406 | - | 0.866 | 0.974 | 0.995 | **1.8** |
| **HR-DiffusionDepth (Ours)** | D | 640 × 192 | **0.078** | **0.351** | **3.120** | **0.120** | **0.925** | **0.986** | **0.996** | 6.26 |

Note: Type D denotes supervised methods, and Type M denotes self-supervised methods. ↓ indicates that a lower value is better, whereas ↑ indicates that a higher value is better. The best result for each metric is shown in bold.

**Table 4.** High-resolution quantitative comparison results on the KITTI dataset.

| Method | Type | Input Resolution | Abs Rel ↓ | Sq Rel ↓ | RMSE ↓ | RMSE log ↓ | $\delta < 1.25$ ↑ | $\delta < 1.25^2$ ↑ | $\delta < 1.25^3$ ↑ | Params (M) |
|---|---|---|---|---|---|---|---|---|---|---|
| 1024 × 320 | | | | | | | | | | |
| Monodepth2 [27] | M | 1024 × 320 | 0.115 | 0.882 | 4.701 | 0.190 | 0.879 | 0.961 | 0.982 | 14.3 |
| R-MSFM3 [52] | M | 1024 × 320 | 0.112 | 0.773 | 4.581 | 0.189 | 0.879 | 0.960 | 0.982 | 3.5 |
| R-MSFM6 [52] | M | 1024 × 320 | 0.108 | 0.748 | 4.470 | 0.185 | 0.889 | 0.963 | 0.982 | 3.8 |
| HR-Depth [44] | M | 1024 × 320 | 0.106 | 0.755 | 4.472 | 0.181 | 0.892 | 0.966 | 0.984 | 14.7 |
| CAD-Depth [45] | M | 1024 × 320 | 0.102 | 0.834 | 4.407 | 0.178 | 0.898 | 0.966 | 0.984 | 18.8 |
| Lite-Mono [28] | M | 1024 × 320 | 0.102 | 0.746 | 4.444 | 0.179 | 0.896 | 0.965 | 0.983 | **3.1** |
| Lite-Mono-8M [28] | M | 1024 × 320 | 0.097 | 0.710 | 4.309 | 0.174 | 0.905 | 0.967 | 0.984 | 8.7 |
| DIFFNet [46] | M | 1024 × 320 | 0.097 | 0.722 | 4.345 | 0.174 | 0.907 | 0.967 | 0.984 | 10.9 |
| MonoViT [9] | M | 1024 × 320 | 0.096 | 0.714 | 4.292 | 0.172 | 0.908 | 0.968 | 0.984 | 27.0 |
| TinyDepth(S) [47] | M | 1024 × 320 | 0.093 | 0.615 | 4.116 | 0.167 | 0.909 | 0.969 | 0.985 | 6.2 |
| RA-Depth [48] | M | 1024 × 320 | 0.092 | 0.634 | 4.135 | 0.168 | 0.912 | 0.969 | 0.985 | 10.0 |
| **HR-DiffusionDepth (Ours)** | D | 1024 × 320 | **0.088** | **0.321** | **2.980** | **0.121** | **0.933** | **0.988** | **0.997** | 6.26 |
| 1280 × 384 | | | | | | | | | | |
| SGDepth [53] | M | 1280 × 384 | 0.107 | 0.768 | 4.468 | 0.186 | 0.891 | 0.963 | 0.982 | - |
| HR-Depth [44] | M | 1280 × 384 | 0.101 | 0.716 | 4.395 | 0.179 | 0.892 | 0.966 | 0.984 | 14.7 |
| CAD-Depth [45] | M | 1280 × 384 | 0.102 | 0.715 | 4.312 | 0.176 | 0.900 | 0.968 | 0.984 | 18.8 |
| MonoViT [9] | M | 1280 × 384 | 0.094 | 0.682 | 4.200 | 0.170 | 0.912 | 0.969 | 0.984 | 27.0 |
| RA-Depth [48] | M | 1280 × 384 | 0.091 | 0.610 | 4.052 | 0.167 | 0.914 | 0.970 | 0.985 | 10.0 |
| TinyDepth(S) [47] | M | 1280 × 384 | 0.091 | 0.607 | 4.043 | 0.165 | 0.915 | 0.970 | 0.985 | 6.2 |
| TuMDE [49] | D | 1280 × 384 | 0.094 | 0.447 | 3.705 | - | 0.893 | 0.977 | 0.995 | 3.4 |
| FastDepth [29] | D | 1280 × 384 | 0.094 | 0.499 | 3.983 | - | 0.889 | 0.974 | 0.993 | 3.9 |
| GuidedDepth [50] | D | 1280 × 384 | 0.115 | 0.736 | 4.227 | - | 0.868 | 0.970 | 0.991 | 5.8 |
| TST [51] | D | 1280 × 384 | 0.087 | 0.437 | 3.798 | - | 0.905 | 0.983 | **0.997** | **1.8** |
| **HR-DiffusionDepth (Ours)** | D | 1280 × 384 | **0.068** | **0.268** | **2.730** | **0.105** | **0.944** | **0.991** | **0.997** | 6.26 |

Note: Type D denotes supervised methods, and Type M denotes self-supervised methods. ↓ indicates that a lower value is better, whereas ↑ indicates that a higher value is better. The best result for each metric is shown in bold.

Furthermore, from the comparison of parameter counts of different methods calculated in Tables 3 and 4, it can be seen that the HR-DiffusionDepth method is highly efficient. For example, RA-Depth and Mono-ViT require 10 M and 27 M parameters

respectively, while HR-DiffusionDepth only needs 6.26 M parameters to achieve better performance. Compared with TinyDepth, the parameter count of the two is comparable, but HR-DiffusionDepth significantly leads in all indicators. Particularly noteworthy is that, under the same premise of using HRNet as the backbone network, the parameter count of HR-DiffusionDepth is approximately 42% of HR-Depth, yet it significantly outperforms the latter in all evaluation indicators, further demonstrating the advancement and efficiency of the proposed architecture.

Figure 3 presents the qualitative comparison results of HR-DiffusionDepth with Lite-Mono and Monodepth2 in terms of capturing local edge details. In Figure 3, the key areas where HR-DiffusionDepth significantly outperforms the comparison methods are marked by dashed boxes. From Figure 3, it can be observed that for targets such as trees, billboards, and traffic signs, this method not only has higher depth accuracy but also can generate clearer and sharper edge structures. Moreover, for distant targets in the scene, HR-DiffusionDepth also demonstrates better resolution capabilities, making the geometric contours of distant targets more distinct. Specifically, in the black box area marked in the first row of Figure 3, the Lite-Mono method estimates that the tree trunk has a significant depth adhesion with the background, while the predicted trunk part by Monodepth2 shows inconsistent depth from top to bottom. In contrast, the edges generated by HR-DiffusionDepth are smooth and consistent, effectively suppressing depth jumps. In the second row of Figure 3, the traffic sign area further validates this trend: Lite-Mono and Monodepth2 both have difficulty clearly separating the sign from the surrounding background, while this method can precisely outline the sign contour. The middle and distant scene shown in the third row of Figure 3 further demonstrates the advantages of HR-DiffusionDepth: Monodepth2 and Lite-Mono failed to accurately estimate the depth of the left tree in the frame, while HR-DiffusionDepth successfully restored the complete geometric information of this target. The above results confirm that HR-DiffusionDepth has a clear advantage in fine edge depth estimation.

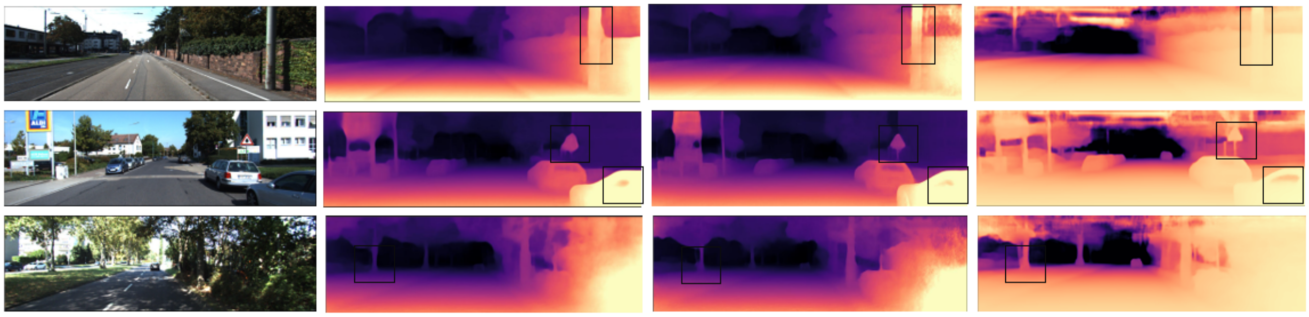


**Figure 3.** Qualitative comparison results on selected samples from the KITTI dataset, illustrating representative differences in local edge sharpness; quantitative results across the full test set are reported in Tables 3 and 4. (**a**) RGB image. (**b**) Lite-Mono. (**c**) Monodepth2. (**d**) HR-DiffusionDepth.

b. Evaluation Results on the SPREAD Dataset

We also trained and evaluated our model on SPREAD and tested the effectiveness of the proposed HR-DiffusionDepth method in tree scene scenarios. Table 5 presents the test results of HR-DiffusionDepth and the existing representative methods on this dataset. From Table 5, it can be seen that HR-DiffusionDepth achieves the lowest values in all four error metrics (Abs Rel, Sq Rel, RMSE, and RMSE log) (reaching 0.3829, 3.4914, 5.4325, 0.4176 respectively), and obtains the highest scores in three accuracy metrics ($\delta < 1.25$, $\delta < 1.25^2$, $\delta < 1.25^3$) (reaching 0.6521, 0.8226, 0.8965 respectively), outperforming the

comparison methods comprehensively. Taking Abs Rel as an example, compared with SimIPU, which achieves the best Abs Rel among the baselines (though its performance on accuracy metrics such as $\delta < 1.25$ is among the lowest in the table), HR-DiffusionDepth reduces the error by 28.8% and reduces the parameter count by 91.9%. In the RMSE metric, this method further achieves a significant 58.3% reduction. Although the parameter count of HR-DiffusionDepth (6.26 M) is slightly higher than that of FastDepth and GuidedDepth, its accuracy advantage is significant, confirming the effectiveness of the model architecture under a limited parameter budget.

**Table 5.** Quantitative comparison results on the SPREAD dataset.

| Method | Abs Rel ↓ | Sq Rel ↓ | RMSE ↓ | RMSE log ↓ | $\delta < 1.25$ ↑ | $\delta < 1.25^2$ ↑ | $\delta < 1.25^3$ ↑ | Params (M) |
|---|---|---|---|---|---|---|---|---|
| FastDepth [29] | 0.749 | 10.845 | 17.728 | 1.611 | 0.028 | 0.044 | 0.0757 | **3.96** |
| GuidedDepth [50] | 0.7674 | 11.1929 | 8.7004 | 0.6196 | 0.436 | 0.7085 | 0.8239 | 5.83 |
| SimIPU_Res50 [54] | 0.5384 | 5.9954 | 13.0391 | 0.8831 | 0.0301 | 0.0962 | 0.3941 | 77.77 |
| **HR-DiffusionDepth (Ours)** | **0.3829** | **3.4914** | **5.4325** | **0.4176** | **0.6521** | **0.8226** | **0.8965** | 6.26 |

Note: ↓ indicates that a lower value is better, whereas ↑ indicates that a higher value is better. The best result for each metric is shown in bold.

Figure 4 presents the qualitative comparison results of HR-DiffusionDepth and FastDepth on the SPREAD tree scene dataset. As shown in Figure 4, HR-DiffusionDepth can generate depth maps with precise numerical values and strong spatial continuity, clearly representing the scene structure and sharpening the target boundaries. Compared to FastDepth, this method demonstrates higher clarity and geometric fidelity in edge depiction. Specifically, in the first row of the example in Figure 4, FastDepth fails to effectively separate the tree canopy area from the background, resulting in the inaccurate distinction of fine structures such as leaves; while HR-DiffusionDepth successfully restores the fine depth differences of leaves, significantly enhancing the visibility of small targets. The second row of the example in Figure 4 further validates this advantage: HR-DiffusionDepth sketched the geometric outline of the fence with smoother and sharper edges, while FastDepth produced blurry boundaries with shape distortions. In summary, the experiments on the SPREAD dataset confirm that HR-DiffusionDepth has strong depth estimation capabilities in complex tree scenes. The high-quality depth maps generated by HR-DiffusionDepth can provide reliable basic data support for downstream tasks such as tree monitoring, resource investigation, and ecological research.

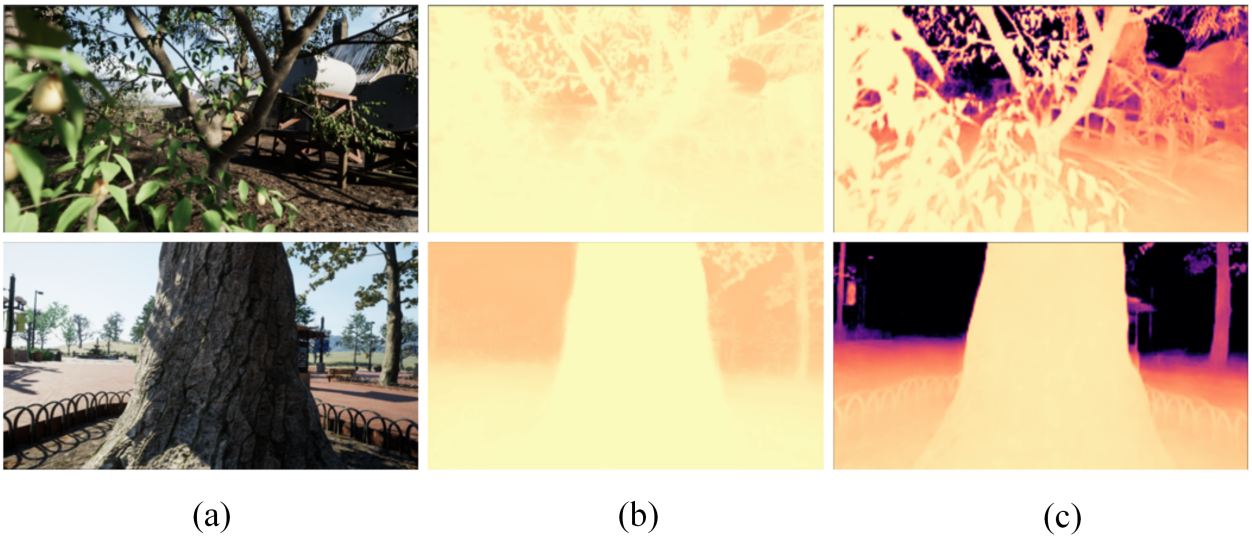


**Figure 4.** Qualitative comparison results on the SPREAD dataset. (**a**) RGB image. (**b**) FastDepth. (**c**) HR-DiffusionDepth.

#### 4.1.2. Model Complexity Analysis

To verify the feasibility of HR-DiffusionDepth in practical applications, this paper conducted inference complexity analysis on NVIDIA Titan Xp and GeForce RTX 3090 GPUs (NVIDIA Corporation, Santa Clara, CA, USA). The test images were taken from the KITTI dataset, and the input resolution was uniformly set to 640 × 192. The model complexity was quantified by the number of floating-point operations (FLOPs) and the single-frame inference time. The results are listed in Table 6.

As shown in Table 6, Monodepth2 had the lowest inference latency, but the depth estimation accuracy was the worst, and Abs Rel ranked at the bottom among all the comparison methods. The FLOPs of HR-DiffusionDepth were the lowest (4.41 G), only 86.5% of that of the suboptimal method Lite-Mono (5.1 G). At the same time, Abs Rel (0.078) was further reduced by 23.5%. Due to the forward noising and multi-step iterative denoising required by the diffusion model during the inference stage, the inference speed of HR-DiffusionDepth was the slowest among the comparison methods. Considering both accuracy and complexity, HR-DiffusionDepth outperformed the existing lightweight methods in terms of computational efficiency and estimation accuracy. Given the multi-second latency reported below, the proposed method is best characterized as a single-shot measurement tool. To verify the feasibility of on-device deployment, we converted HR-DiffusionDepth to Open Neural Network Exchange (ONNX) format and deployed it on a realme GT Neo smartphone (realme, Shenzhen, China) running Android 13. The size of the converted model was 26.7 MB. Using ONNX Runtime 1.27.0 for inference, the single-frame latency was 7261 ms, and the runtime memory usage was 187 MB. In addition, we measured the complete end-to-end latency on this device, covering image preprocessing, trunk segmentation (via the SAM API), depth estimation, and DBH calculation. The total latency per frame was 9125 ms. During this process, the processor utilization was approximately 72%, and the power consumption was approximately 4.6 W. Among these steps, the trunk segmentation part is completed by calling the cloud-based SAM API. This step depends on network connectivity.

**Table 6.** Comparison of model complexity across different methods.

| Method | FLOPs (G) | Abs Rel ↓ | $\delta < 1.25$ ↑ | Time (ms) Titan Xp | Time (ms) RTX 3090 |
|---|---|---|---|---|---|
| Monodepth2 [27] | 8.0 | 0.115 | 0.879 | **7.4** | **7.5** |
| R-MSFM3 [52] | 16.5 | 0.112 | 0.879 | 13.2 | 11.7 |
| R-MSFM6 [52] | 31.2 | 0.108 | 0.889 | 22.5 | 18.4 |
| Lite-Mono [28] | 5.1 | 0.102 | 0.896 | 13.7 | 12.1 |
| **HR-DiffusionDepth (Ours)** | **4.41** | **0.078** | **0.925** | 60.0 | 40.2 |

Note: FLOPs are in GigaFLOPs, and inference time is measured on Titan Xp and RTX 3090 GPUs. ↓ indicates that a lower value is better, whereas ↑ indicates that a higher value is better. The best result for each metric is shown in bold.

#### 4.1.3. Comparison with Representative Heavyweight Methods

In a lightweight design, some precision is inevitably lost. However, by comparing it with some heavyweight representative methods to further verify the applicability of our proposed HR-DiffusionDepth method in field operations, the results are shown in Table 7.

In terms of error and accuracy indicators, our method even surpasses the Yin method. It is 0.004 lower in Abs Rel and 0.528 lower in RMSE. Moreover, compared with DepthFormer, our parameter count is 2.2% of its, but the value of the $\delta < 1.25$ indicator only differs by 0.03 from DepthFormer, which is 97% of DepthFormer's. For $\delta < 1.25^3$, it even reaches 99% of DepthFormer's. Although our method is less accurate than these models, our computational cost is much lower than these methods. As shown in Figure 5, we calculated the ratio of depth accuracy ($\delta < 1.25$) to the parameter count. The larger the

ratio, the more successful the lightweight design (that is, a relatively small computational cost achieves a higher accuracy).

**Table 7.** Comparison results with heavyweight methods on the KITTI dataset.

| Method | Abs Rel ↓ | Sq Rel ↓ | RMSE ↓ | RMSE log ↓ | $\delta < 1.25$ ↑ | $\delta < 1.25^2$ ↑ | $\delta < 1.25^3$ ↑ | Params (M) | $\delta < 1.25$/ Params ↑ |
|---|---|---|---|---|---|---|---|---|---|
| Eigen [8] | 0.203 | 1.548 | 6.307 | 0.282 | 0.702 | 0.898 | 0.967 | 141 | 0.005 |
| Yin [55] | 0.072 | - | 3.258 | 0.117 | 0.938 | 0.990 | 0.998 | 114 | 0.008 |
| BTS [56] | 0.059 | 0.245 | 2.756 | 0.096 | 0.956 | 0.993 | 0.998 | 47 | 0.020 |
| Adabins [57] | 0.058 | 0.190 | 2.360 | 0.088 | 0.964 | 0.995 | **0.999** | 78 | 0.012 |
| DPT-Hybrid [58] | 0.062 | - | 2.573 | 0.092 | 0.959 | 0.995 | **0.999** | 112 | 0.008 |
| DepthFormer (Agarwal 2022) [59] | 0.058 | 0.187 | 2.285 | 0.087 | 0.967 | 0.996 | **0.999** | 254 | 0.004 |
| D-Net [60] | 0.056 | 0.189 | 2.362 | 0.087 | 0.963 | 0.995 | **0.999** | 358 | 0.003 |
| DepthFormer (Li 2023) [61] | 0.052 | 0.158 | 2.143 | **0.079** | 0.975 | **0.997** | **0.999** | 279 | 0.004 |
| eViTBins [62] | **0.051** | **0.149** | **2.126** | 0.081 | **0.976** | **0.997** | **0.999** | 98 | 0.010 |
| **HR-DiffusionDepth (Ours)** | 0.068 | 0.268 | 2.730 | 0.105 | 0.944 | 0.991 | 0.997 | **6.26** | **0.151** |

Note: Depth errors (↓) and depth accuracy (↑) are reported. ↓ indicates that a lower value is better, whereas ↑ indicates that a higher value is better. The best result for each metric is shown in bold.

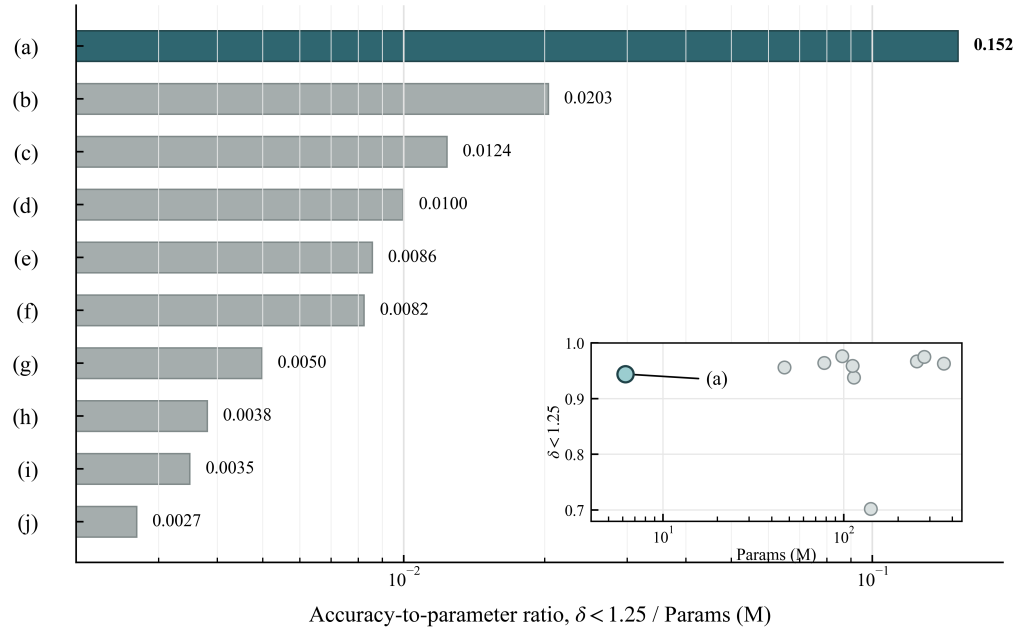


**Figure 5.** Ratio of accuracy to parameter count for different models. (a) HR-DiffusionDepth. (b) BTS. (c) AdaBins. (d) eViTBins. (e) DPT-Hybrid. (f) Yin. (g) Eigen. (h) DepthFormer ([59]). (i) DepthFormer ([61]). (j) D-Net.

From Figure 5, it can be seen that the HR-DiffusionDepth method is far superior to other methods. This confirms the success of our lightweight design and also indicates that the HR-DiffusionDepth method has higher computational efficiency and lower computational cost and is suitable for mobile deployment. We further examined the trade-off relationship between accuracy ($\delta < 1.25$) and parameter count of the lightweight methods in Table 3, and presented it in Figure 6 in the form of a Pareto frontier analysis.

Three methods are located on this frontier: TST (1.8 M parameters, $\delta < 1.25 = 0.866$), Lite-Mono (3.1 M parameters, $\delta < 1.25 = 0.886$), and HR-DiffusionDepth (6.26 M parameters, $\delta < 1.25 = 0.925$). Among them, HR-DiffusionDepth achieves the highest accuracy, and its parameter count remains at the single-digit-million level, on the same order of magnitude as the other lightweight methods on the frontier.

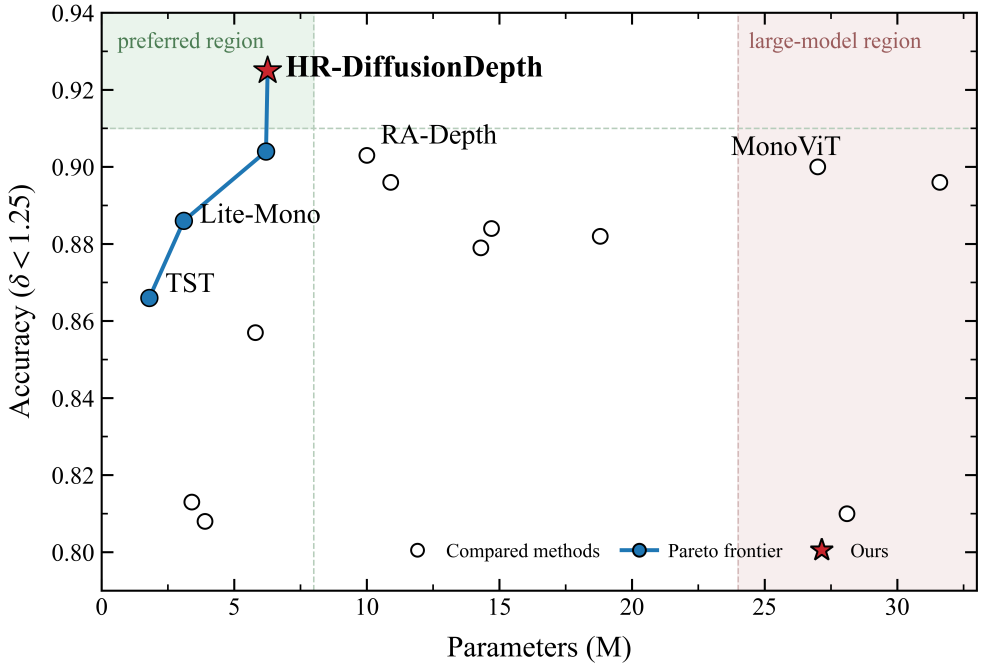


**Figure 6.** Pareto frontier analysis of accuracy ($\delta < 1.25$) versus parameter count among lightweight monocular depth estimation methods on the KITTI dataset (640 × 192 resolution). The dashed lines indicate the accuracy and parameter-count thresholds used to define the preferred region and the large-model region.

#### 4.1.4. The Role of Depth Information

Obtaining depth information is the most important step in the MobileDBH proposed in this paper, and it is also an important technical approach to reduce the cost of expensive 3D equipment. To quantitatively evaluate the impact of depth information on the accuracy of DBH estimation, this study designed a control experiment, dividing the input modalities into two categories: (i) only RGB images and (ii) RGB images from the same perspective along with their corresponding depth maps. The evaluation indicators selected were the mean absolute error (MAE) and the root mean square error (RMSE), both of which strictly measure the systematic deviation between the predicted values and the actual values. The experimental results are shown in Table 8.

**Table 8.** DBH estimation accuracy with and without depth information.

| Method | RMSE (cm) | MAE (cm) |
|---|---|---|
| RGB only | 5.34 | 4.72 |
| RGB + depth | 3.10 | 2.25 |

As can be seen from Table 8, when using only RGB input, the RMSE and MAE were 5.34 cm and 4.72 cm respectively. When depth information was introduced, these two indicators significantly decreased, reaching 3.10 cm and 2.25 cm respectively. This indicates that due to the lack of depth information, DBH estimation can only be performed at the two-dimensional level, and tree trunks often exhibit tilted or curved growth. RGB images can only observe the two-dimensional projection and cannot perceive these three-dimensional deformations, resulting in larger errors. This result confirms that depth information is essential in estimating DBH, and adding depth information can significantly improve the accuracy of DBH estimation. Missing depth information and relying solely on two-dimensional plane mapping will bring substantial systematic errors.

### *4.2. Ablation Study*

In this section, we performed ablation studies on the KITTI dataset to validate the contribution of individual components within HR-DiffusionDepth. The goal is to quantitatively assess how each module and key parameter affects the overall depth estimation performance.

#### 4.2.1. Module Effectiveness

The feature extraction module of HR-DiffusionDepth is primarily composed of two key components: the EdgeModule and the MultiScaleModule. To assess their individual impacts, we performed an ablation study on the KITTI dataset with four distinct configurations: (1) baseline (RGB input only); (2) baseline + MultiScaleModule; (3) baseline + EdgeModule; and (4) the complete HR-DiffusionDepth model, which integrates both modules. The corresponding results are summarized in Table 9.

**Table 9.** Ablation study results of HR-DiffusionDepth.

| Model | EdgeModule | MultiScaleModule | Abs Rel ↓ | RMSE ↓ | $\delta < 1.25$ ↑ | Edge Gradient Consistency ↑ | Contour F-Score ↑ | FLOPs (G) |
|---|---|---|---|---|---|---|---|---|
| Baseline | × | × | 0.323 | 7.031 | 0.546 | 0.312 | 0.286 | **0.17** |
| Baseline + EdgeModule | ✓ | × | 0.218 | 5.720 | 0.663 | 0.423 | 0.397 | 0.19 |
| Baseline + MultiScaleModule | × | ✓ | **0.0780 ± 0.0026** | 3.195 ± 0.039 | **0.9243 ± 0.0012** | 0.568 | 0.542 | 3.66 |
| **HR-DiffusionDepth (Ours)** | ✓ | ✓ | 0.0783 ± 0.0015 | **3.126 ± 0.029** | **0.9243 ± 0.0031** | **0.632** | **0.604** | 4.41 |

Note: The symbols ✓ and × indicate the presence or absence of a module, respectively. ↓ indicates that a lower value is better, whereas ↑ indicates that a higher value is better. The best result for each metric is shown in bold.

As presented in Table 9, the baseline, devoid of both edge and multi-scale information, yields suboptimal accuracy (Abs Rel: 0.323, $\delta < 1.25$: 0.546). A significant improvement is observed with the MultiScaleModule alone, which achieves a 76% relative reduction in Abs Rel. Similarly, the EdgeModule in isolation also contributes substantially, lowering the Abs Rel by 33%. The complete HR-DiffusionDepth model obtains the best RMSE and matches the highest $\delta < 1.25$ accuracy. This result is visually supported by Figure 7, where the model with the EdgeModule produces sharper object boundaries and more distinct long-range structures. In the complete model, the effect of the EdgeModule is also reflected in a numerical improvement of the RMSE metric, from $3.195 \pm 0.039$ to $3.126 \pm 0.029$, with a corresponding increase in FLOPs of approximately 20% (from 3.66 G to 4.41 G). Since RMSE is more sensitive to a small number of pixels with larger errors than Abs Rel, which is the linear average of errors across all pixels in the image, such localized improvements may not be clearly reflected in the two full-image average metrics, Abs Rel and $\delta < 1.25$. Abs Rel changed from $0.0780 \pm 0.0026$ to $0.0783 \pm 0.0015$, while $\delta < 1.25$ remained at 0.9243; see Appendix A Table A1 for per-run results. Besides RMSE, we also computed two edge-specific metrics, namely Edge Gradient Consistency and Contour F-score, both evaluated against the ground-truth object boundaries. The complete model achieves values of 0.632 and 0.604 on these two metrics, respectively, compared with 0.568 and 0.542 for the configuration using only the MultiScaleModule. We conducted ablation experiments of the EdgeModule on the SPREAD forest dataset, comparing the configuration using only MultiScaleModule with the complete model. The results are shown in Table 10. After adding the EdgeModule, Abs Rel decreased from 0.3982 to 0.3829, RMSE decreased from 5.6770 to 5.4325, and all other indicators also improved. Since DBH estimation is highly dependent on the positioning of the tree trunk boundary, the aforementioned improvement at the boundary level is closely related to this task. This supports the design choice of retaining the EdgeModule despite the additional computational cost. We also consider making EdgeModule an optional component in future implementations, allowing it to be enabled or disabled flexibly based on the available computational budget.

**Table 10.** Ablation study results of the EdgeModule on the SPREAD dataset.

| Model | Abs Rel ↓ | Sq Rel ↓ | RMSE ↓ | RMSE log ↓ | $\delta < 1.25$ ↑ | $\delta < 1.25^2$ ↑ | $\delta < 1.25^3$ ↑ |
|---|---|---|---|---|---|---|---|
| SPREAD: MultiScale-only | 0.3982 | 3.7707 | 5.6770 | 0.4377 | 0.6371 | 0.8136 | 0.8915 |
| **SPREAD: HR-DiffusionDepth (Ours)** | **0.3829** | **3.4914** | **5.4325** | **0.4176** | **0.6521** | **0.8226** | **0.8965** |

Note: ↓ indicates that a lower value is better, whereas ↑ indicates that a higher value is better. The best result for each metric is shown in bold.

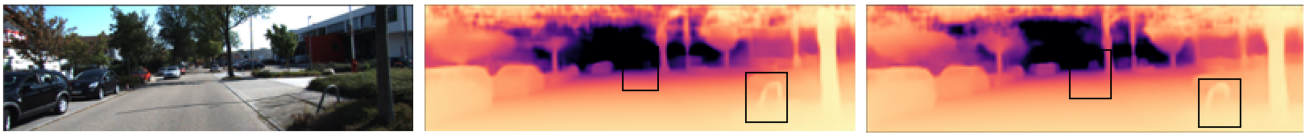


**Figure 7.** Qualitative results of the ablation study on the EdgeModule. (**a**) RGB image. (**b**) Model without the EdgeModule. (**c**) Model with the EdgeModule. The black boxes highlight representative regions for detailed comparison.

#### 4.2.2. Denoising Steps

To investigate the impact of the number of diffusion inference steps on the model's accuracy, we evaluated the model performance under different denoising steps. The results are shown in Table 11 and Figure 8. As shown in Table 11, when only single-step sampling is used, HR-DiffusionDepth can achieve the optimal accuracy while maintaining the minimum FLOPs (4.41 G) and the shortest inference time (40.2 ms). As shown in Figure 8, as the number of steps increases, the model accuracy decreases, while both FLOPs and inference time increase. These results fully demonstrate that this model can achieve the best inference performance with the lowest computational cost, further verifying the efficiency of HR-DiffusionDepth as a single-shot measurement tool.

**Table 11.** Quantitative results with varying denoising steps.

| Step | Abs Rel ↓ | RMSE ↓ | $\delta < 1.25$ ↑ | FLOPs (G) | Time (ms) |
|---|---|---|---|---|---|
| 1 | **0.078** | **3.120** | **0.925** | **4.41** | **40.2** |
| 2 | 0.080 | 3.187 | 0.923 | 5.48 | 47.9 |
| 3 | 0.080 | 3.181 | 0.922 | 6.55 | 41.6 |
| 4 | 0.079 | 3.160 | 0.924 | 7.62 | 40.5 |

Note: ↓ indicates that a lower value is better, whereas ↑ indicates that a higher value is better. The best result for each metric is shown in bold.

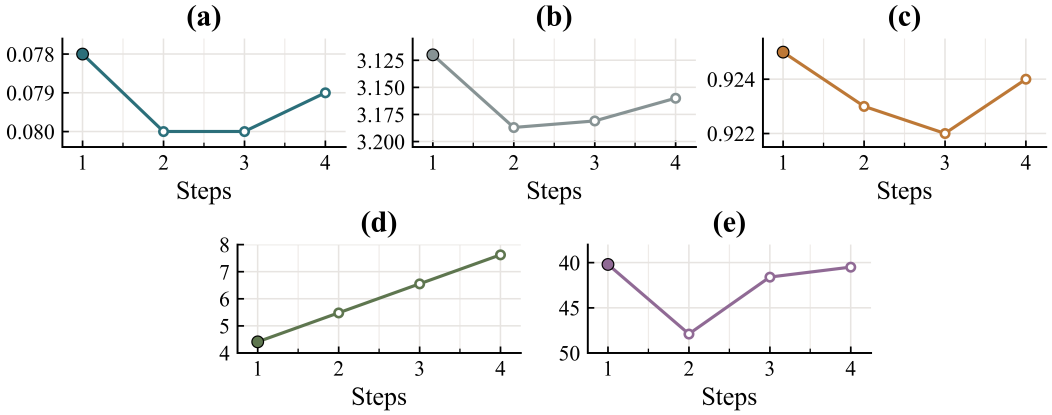


**Figure 8.** Performance trends with varying denoising steps. (**a**) Abs Rel. (**b**) RMSE. (**c**) $\delta < 1.25$. (**d**) FLOPs. (**e**) Time.

#### 4.2.3. Convolutional Network Depth

In order to investigate the impact of the depth of the denoising network on the accuracy, we evaluated the performance of the models under different configurations of convolutional layers. As shown in Table 12 and Figure 9, when the number of convolution layers is reduced to 2, the model accuracy reaches the optimal level. When the network is further deepened, the Abs Rel remains stable, while the RMSE first increases and then decreases. There is no further improvement in overall performance. Therefore, the final design adopts a 2-layer convolution, with corresponding parameters amounting to only 0.071 M and FLOPs being only 0.545 G. We further constructed a control model that shares the same backbone and conditional features with the original model. Only the denoising module is replaced with a regression decoder (without the noise addition and iterative denoising process, directly outputting the depth map). The results, reported as mean ± SD over three runs, are shown in Table 13; see Appendix A Table A2 for per-run results. Under the condition that the parameter count (6.18 M compared with 6.26 M) and FLOPs (4.26 G compared with 4.41 G) are similar, the Abs Rel of HR-DiffusionDepth is $0.0783 \pm 0.0015$, the RMSE is $3.126 \pm 0.029$, and $\delta < 1.25$ is $0.9243 \pm 0.0031$; the corresponding values of the control model are $0.0863 \pm 0.0025$, $3.431 \pm 0.040$, and $0.9157 \pm 0.0025$.

**Table 12.** Quantitative results of altering denoising convolutional layers.

| Layers | Abs Rel ↓ | RMSE ↓ | $\delta < 1.25$ ↑ | Params (M) | FLOPs (G) |
|---|---|---|---|---|---|
| 2 | **0.078** | **3.120** | **0.925** | **0.071** | **0.545** |
| 4 | 0.080 | 3.213 | 0.921 | 0.089 | 0.691 |
| 6 | 0.080 | 3.192 | 0.920 | 0.095 | 0.733 |

Note: ↓ indicates that a lower value is better, whereas ↑ indicates that a higher value is better. The best result for each metric is shown in bold.

**Table 13.** Comparison between HR-DiffusionDepth and a non-diffusion regression decoder.

| Model | Abs Rel ↓ | RMSE ↓ | $\delta < 1.25$ ↑ | Params (M) | FLOPs (G) |
|---|---|---|---|---|---|
| Plain regression decoder | $0.0863 \pm 0.0025$ | $3.431 \pm 0.040$ | $0.9157 \pm 0.0025$ | **6.180** | **4.260** |
| **HR-DiffusionDepth (Ours)** | $\mathbf{0.0783 \pm 0.0015}$ | $\mathbf{3.126 \pm 0.029}$ | $\mathbf{0.9243 \pm 0.0031}$ | **6.260** | **4.410** |

Note: ↓ indicates that a lower value is better, whereas ↑ indicates that a higher value is better. The best result for each metric is shown in bold.

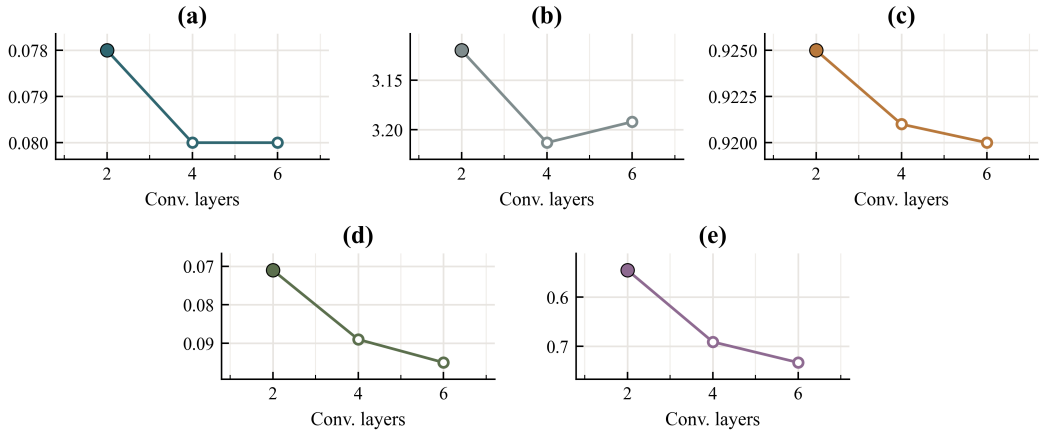


**Figure 9.** Performance trends when altering denoising convolutional layers. (**a**) Abs Rel. (**b**) RMSE. (**c**) $\delta < 1.25$. (**d**) Params. (**e**) FLOPs.

### *4.3. Case Study for DBH Estimation*

To evaluate the feasibility and accuracy of MobileDBH in real scenarios, we developed a corresponding DBH estimation app based on the proposed HR-DiffusionDepth model

(see Figure 10), and collected multiple images from Beijing Forestry University to form a small field test set, simulating the real field investigation work environment. Specifically, 20 representative trees were selected, and the trunk circumference at 1.3 m was measured and converted to DBH values (diameter = circumference/$\pi$), covering a DBH range of 11.46–30.86 cm.

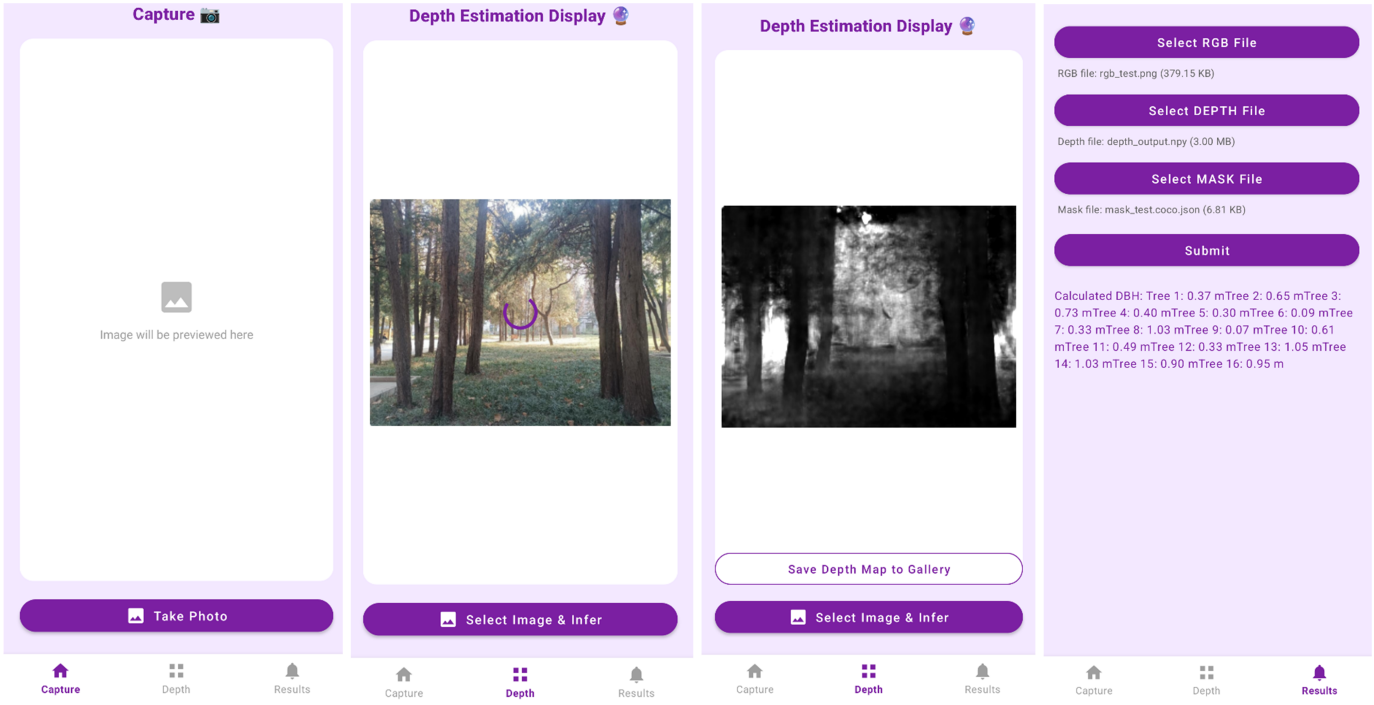


**Figure 10.** Complete user workflow for DBH estimation using the app. (**a**,**b**) Input RGB images. (**c**) Depth map display interface, showing the depth map generated on-device by HR-DiffusionDepth. (**d**) Result display interface; the trunk segmentation mask is obtained via a call to the Segment Anything Model (SAM) API, while the depth map shown here is the same on-device inference result as in (**c**).

The DBH values estimated by the application were compared with the manually measured values, and the results are summarized in Figure 11, Tables 14 and A3. Figure 11 shows a strong linear correlation between the predicted values and the measured values. Among the 20 field samples, Tree 5 showed the greatest deviation between the predicted value and the actual measurement (see Appendix A Table A3). The measurement results for all 20 trees were as follows: RMSE = 3.10 cm, MAE = 2.25 cm. In addition, the Holcomb method (an existing method based on mobile devices) was used as a benchmark for comparison. Both methods were scored based on the same 20 trees and the same on-site photos. The RMSE of HR-DiffusionDepth is lower than that of the Holcomb method (3.10 cm compared with 3.70 cm), and the MAE of the two methods is comparable (2.25 cm compared with 2.20 cm). This difference is smaller than the measurement error inherent in the tape measure itself and the error usually caused by converting the circumference to the diameter of an irregular tree trunk by dividing by $\pi$.

**Table 14.** Comparison results of different DBH methods.

| Method | RMSE (cm) | MAE (cm) |
|---|---|---|
| Holcomb [5] | 3.70 | **2.20** |
| **Our method** | **3.10** | 2.25 |

Note: The best result for each metric is shown in bold.

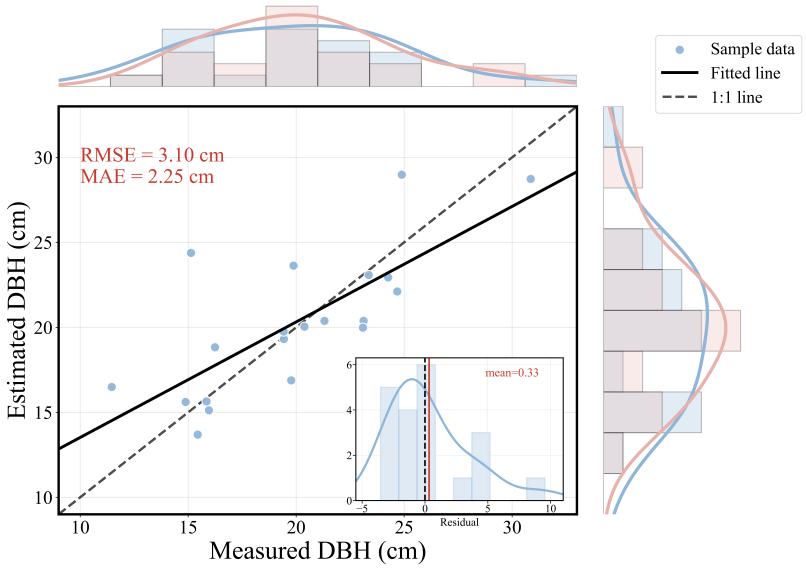


**Figure 11.** Correlation between estimated DBH and measured DBH.The colored curves and bars along the top and right margins show the distributions of measured and estimated DBH values. The inset shows the distribution of residuals, and the red vertical line indicates the mean residual.

## 5. Discussion

The experimental results show that HR-DiffusionDepth provides a favorable balance between depth accuracy and computational cost. Compared with existing lightweight methods, its advantage is not only reflected in lower Abs Rel and RMSE but also in the fact that these results are obtained with a compact parameter scale. When sharing the same HRNet backbone, HR-DiffusionDepth achieves competitive or superior performance with only 42% of the parameters of HR-Depth. This indicates that replacing the heavy encoder and fusion structure with the proposed lightweight multi-scale design is an effective strategy for mobile-oriented monocular depth estimation. Furthermore, we applied the SPREAD-trained HR-DiffusionDepth model in a zero-shot manner to 101 real images from the publicly available CanaTree100 dataset [19] (see Appendix A Table A4) to evaluate depth estimation accuracy. On the SPREAD dataset, HR-DiffusionDepth achieves an Abs Rel of 0.3829 and an RMSE of 5.43 m; on CanaTree100, Abs Rel is 0.468 and $\delta < 1.25$ is 0.512, meaning that approximately half of the pixels fall within 25% of the ground-truth depth.

The qualitative results further explain why the proposed framework is suitable for DBH estimation. In both road-scene and forest-scene examples, the main advantage of HR-DiffusionDepth appears around object boundaries, where competing methods often produce depth bleeding, blurred contours, or incomplete separation between foreground and background. This is particularly important for tree measurement because DBH estimation depends on accurately locating the trunk boundary and estimating its spatial scale. The ablation study also supports this interpretation: the MultiScaleModule provides the main performance gain by improving global and contextual depth representation, while the EdgeModule further reduces large boundary-related errors and produces sharper depth

maps. Therefore, the combination of multi-scale context and edge cues is more appropriate for tree measurement scenes than relying on either component alone.

The DBH ablation experiment confirms that depth information is essential rather than auxiliary in the proposed mobile measurement pipeline. RGB-only estimation is limited to a 2D projection of the trunk and cannot adequately account for three-dimensional trunk deformation, camera distance, or local surface geometry. By incorporating monocular depth, MobileDBH reduces RMSE from 5.34 cm to 3.10 cm and MAE from 4.72 cm to 2.25 cm. This improvement explains the practical value of replacing expensive 3D equipment with a smartphone-based depth estimation model: the system retains the low-cost and convenient acquisition mode of RGB imagery while recovering spatial information needed for reliable DBH estimation.

The lightweight design also has practical implications for deployment. Although diffusion-based models often require iterative denoising and therefore introduce additional latency, our experiments show that one denoising step is sufficient for the best overall performance. Increasing the number of steps does not improve accuracy and instead increases FLOPs and inference time. Similarly, deepening the denoising network beyond two convolutional layers brings no clear benefit. These findings suggest that, in this task, strong conditional features from the feature extraction and fusion modules are more important than a deeper or longer denoising process. This observation is consistent with the goal of mobile deployment, where unnecessary denoising iterations should be avoided.

The in situ app test demonstrates that the proposed method can be integrated into an operational DBH measurement workflow. The results on all 20 field samples show a lower RMSE than the Holcomb method, while the two methods achieve a comparable MAE, supporting the feasibility of smartphone-based DBH estimation. Nevertheless, the field experiment remains limited in scale and site diversity. We manually annotated the ground-truth trunk masks for these 20 images and compared them with the corresponding SAM segmentation results. The obtained average IoU was 0.8507, and the average Dice coefficient was 0.9086. The per-tree segmentation results are provided in Appendix A Table A5. This suggests that the end-to-end measurement accuracy still depends on the joint reliability of depth estimation, trunk segmentation, camera calibration, and user image acquisition.

In future work, the introduction of Explainable Artificial Intelligence (XAI) techniques is expected to help identify the specific sources of end-to-end DBH measurement errors. For instance, visualizing the degree of attention paid by the depth estimation model to the trunk boundary area, or analyzing the contribution ratios of depth information, segmentation masks, and scale calibration to the final DBH prediction, can help diagnose abnormal measurement cases more accurately and provide a basis for improving the model's robustness. Additionally, more rigorous quantitative evaluations can be conducted on calibration accuracy (such as reprojection error), and field verification can be extended to a wider range of tree species, lighting conditions, shooting distances, and angles. At the same time, parallel comparisons can be made with traditional tape measure measurements and consumer-grade LiDAR devices. Exploring lightweight and on-device segmentation solutions to replace the current cloud-based segmentation solution is one of the directions for future work. For instance, MobileSAM [63] employs a lightweight TinyViT encoder with approximately 5.78 million parameters, replacing the original ViT-H image encoder of SAM (with approximately 632 million parameters), reducing the encoder parameter count by approximately 100 times.

## 6. Conclusions

This paper presents HR-DiffusionDepth, a lightweight depth estimation network for mobile devices. Based on this model, we developed the MobileDBH application for

estimating tree diameters, which requires a single ruler calibration per measurement scene, after which trees within that scene can be measured simply by taking a photo with a mobile phone. Compared with manual measurement and LiDAR methods, this method has clear advantages in terms of hardware cost and operational convenience. Experiments on multiple datasets show that HR-DiffusionDepth achieves competitive depth estimation accuracy. In the field DBH estimation test, the MobileDBH system achieved a lower RMSE than the Holcomb baseline. The proposed DBH estimation method based on monocular vision shows promising potential in agricultural scenarios such as orchards and plantations.

**Author Contributions:** Conceptualization, C.M., J.F., A.L. and H.Z.; methodology, C.M., J.F. and A.L.; software, A.L., J.F., C.L., K.H., H.D. and J.L.; formal analysis, C.M., J.F. and A.L.; data curation, A.L. and C.L.; writing—original draft preparation, C.M., J.F., A.L. and H.Z.; writing—review and editing, C.M., J.F., A.L., K.H., H.D., J.L. and H.Z.; visualization, A.L. and J.F.; project administration, C.M. and H.Z.; funding acquisition, C.M. and H.Z. All authors have read and agreed to the published version of the manuscript.

**Funding:** This work was supported by the Emergency Open Competition Project of the National Forestry and Grassland Administration (Grant No. 202303), the Outstanding Youth Team Project of Central Universities (Grant No. QNTD202504), and the Beijing Forestry University National Training Program of Innovation and Entrepreneurship for Undergraduates, under the projects entitled "ShanYin: A Lightweight Interpretable Recommendation System Based on Dietary Image Sequences" and "Research and Application of a Clinical Reasoning Large Language Model Based on Few-Shot Learning and Traceability."

**Institutional Review Board Statement:** Not applicable.

**Informed Consent Statement:** Not applicable.

**Data Availability Statement:** The original data presented in the study are openly available in GitHub at https://github.com/ForestryIIP/MobileDBH/tree/main (accessed on 31 July 2026). The KITTI dataset is publicly available at https://www.cvlibs.net/datasets/kitti/ (accessed on 31 July 2026). The SPREAD dataset is publicly available in Zenodo at https://zenodo.org/records/14515915 (accessed on 31 July 2026).

**Acknowledgments:** The authors would like to thank the staff of the Experimental Forest of Beijing Forestry University for their assistance with field data collection, and the anonymous reviewers for their valuable comments and suggestions.

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

## Appendix A. Detailed DBH Estimation Results

**Table A1.** Per-run results of the EdgeModule ablation over three independent runs.

| Model | Run | Abs Rel ↓ | RMSE ↓ | $\delta < 1.25$ ↑ |
|---|---|---|---|---|
| MultiScale-only | 1 | 0.077 | 3.185 | 0.925 |
| MultiScale-only | 2 | 0.081 | 3.238 | 0.923 |
| MultiScale-only | 3 | 0.076 | 3.162 | 0.925 |
| MultiScale-only | Mean | $0.0780 \pm 0.0026$ | $3.195 \pm 0.039$ | $0.9243 \pm 0.0012$ |
| Complete model (MultiScale + Edge) | 1 | 0.078 | 3.120 | 0.925 |
| Complete model (MultiScale + Edge) | 2 | 0.080 | 3.158 | 0.921 |
| Complete model (MultiScale + Edge) | 3 | 0.077 | 3.101 | 0.927 |
| Complete model (MultiScale + Edge) | Mean | $0.0783 \pm 0.0015$ | $3.126 \pm 0.029$ | $0.9243 \pm 0.0031$ |

Note: ↓ indicates that a lower value is better, whereas ↑ indicates that a higher value is better.

**Table A2.** Per-run results of the non-diffusion regression decoder comparison over three independent runs.

| Model | Run | Abs Rel ↓ | RMSE ↓ | $\delta < 1.25$ ↑ |
|---|---|---|---|---|
| Plain regression decoder | 1 | 0.086 | 3.430 | 0.916 |
| Plain regression decoder | 2 | 0.089 | 3.471 | 0.913 |
| Plain regression decoder | 3 | 0.084 | 3.392 | 0.918 |
| Plain regression decoder | Mean | $0.0863 \pm 0.0025$ | $3.431 \pm 0.040$ | $0.9157 \pm 0.0025$ |
| HR-DiffusionDepth (Ours) | 1 | 0.078 | 3.120 | 0.925 |
| HR-DiffusionDepth (Ours) | 2 | 0.080 | 3.158 | 0.921 |
| HR-DiffusionDepth (Ours) | 3 | 0.077 | 3.101 | 0.927 |
| HR-DiffusionDepth (Ours) | Mean | $0.0783 \pm 0.0015$ | $3.126 \pm 0.029$ | $0.9243 \pm 0.0031$ |

Note: ↓ indicates that a lower value is better, whereas ↑ indicates that a higher value is better.

**Table A3.** Field measurement results for the 20 field trees.

| Tree ID | Measured DBH (cm) | Predicted DBH (cm) | Tree ID | Measured DBH (cm) | Predicted DBH (cm) |
|---|---|---|---|---|---|
| 1 | 23.12 | 20.38 | 11 | 19.42 | 19.75 |
| 2 | 23.09 | 19.98 | 12 | 19.77 | 16.88 |
| 3 | 16.24 | 18.83 | 13 | 19.88 | 23.63 |
| 4 | 24.26 | 22.94 | 14 | 15.85 | 15.64 |
| 5 | 15.13 | 24.38 | 15 | 30.86 | 28.73 |
| 6 | 24.68 | 22.11 | 16 | 23.36 | 23.08 |
| 7 | 19.43 | 19.31 | 17 | 15.96 | 15.12 |
| 8 | 11.46 | 16.49 | 18 | 24.89 | 28.98 |
| 9 | 20.38 | 20.04 | 19 | 15.44 | 13.69 |
| 10 | 14.88 | 15.61 | 20 | 21.31 | 20.38 |

**Table A4.** Comparison between the SPREAD test set and the CanaTree100 real-forest dataset (zero-shot).

| Metric | SPREAD Test Set | CanaTree100 (Real Forest, Zero-Shot) |
|---|---|---|
| Number of samples | 1800 | 101 |
| Abs Rel | 0.3829 | 0.468 |
| RMSE (m) | 5.4325 | 5.917 |
| $\delta < 1.25$ | 0.6521 | 0.512 |
| $\delta < 1.25^2$ | 0.8226 | 0.728 |
| $\delta < 1.25^3$ | 0.8965 | 0.848 |

**Table A5.** Per-tree SAM trunk segmentation performance for the 20 field trees.

| Tree ID | IoU | Dice | Tree ID | IoU | Dice |
|---|---|---|---|---|---|
| 1 | 0.6509 | 0.7886 | 11 | 0.9418 | 0.9700 |
| 2 | 0.8984 | 0.9465 | 12 | 0.9036 | 0.9494 |
| 3 | 0.9347 | 0.9662 | 13 | 0.9432 | 0.9708 |
| 4 | 0.9528 | 0.9758 | 14 | 0.7689 | 0.8694 |
| 5 | 0.9625 | 0.9809 | 15 | 0.9052 | 0.9502 |
| 6 | 0.3956 | 0.5669 | 16 | 0.8867 | 0.9399 |
| 7 | 0.9643 | 0.9818 | 17 | 0.4284 | 0.5998 |
| 8 | 0.9704 | 0.9850 | 18 | 0.9496 | 0.9742 |
| 9 | 0.9407 | 0.9694 | 19 | 0.7513 | 0.8580 |
| 10 | 0.9698 | 0.9847 | 20 | 0.8942 | 0.9441 |

## References

1. Wang, J.; Wang, Y.; Zhang, Z.; Wang, W.; Jiang, L. Enhanced awareness of height-diameter allometry in response to climate, soil, and competition in secondary forests. *For. Ecol. Manag.* **2023**, *548*, 121386. https://doi.org/10.1016/j.foreco.2023.121386.
2. Bhebhe, Z.M.; Liu, X.; Zhang, Z.; Paudyal, D.R. Estimation of tree diameter at breast height (DBH) and biomass from allometric models using LiDAR data: A case of the Lake Broadwater Forest in southeast Queensland, Australia. *Remote Sens.* **2025**, *17*, 2523. https://doi.org/10.3390/rs17142523.

3. Calders, K.; Adams, J.; Armston, J.; Bartholomeus, H.; Bauwens, S.; Bentley, L.P.; Chave, J.; Danson, F.M.; Demol, M.; Disney, M.; et al. Terrestrial laser scanning in forest ecology: Expanding the horizon. *Remote Sens. Environ.* **2020**, *251*, 112102. https://doi.org/10.18174/327787.
4. Xuan, J.; Li, X.; Mao, F.; Wang, J.; Zhang, B.; Gong, Y.; Huang, Z.; Chen, C.; Lv, L.; Zhao, Y.; et al. Deep learning combined with smartphone for intelligent estimation of diameter at breast height and AGB in urban forests-dual LiDAR system for validation. *Urban For. Urban Green.* **2025**, *113*, 129119. https://doi.org/10.1016/j.ufug.2025.129119.
5. Holcomb, A.; Tong, L.; Keshav, S. Robust single-image tree diameter estimation with mobile phones. *Remote Sens.* **2023**, *15*, 772. https://doi.org/10.3390/rs15030772.
6. Zhang, J.; Wu, Y.; Jiang, H. Survey on monocular metric depth estimation. *Computers* **2025**, *14*, 502. https://doi.org/10.3390/computers14110502.
7. Fu, K.; Yue, S.; Yin, B. DBH Extraction of Standing Trees Based on a Binocular Vision Method. In *Proceedings of the 2023 IEEE International Instrumentation and Measurement Technology Conference (I2MTC)*; IEEE: Piscataway, NJ, USA, 2023; pp. 1–6. https://doi.org/10.1109/i2mtc53148.2023.10176070.
8. Eigen, D.; Puhrsch, C.; Fergus, R. Depth map prediction from a single image using a multi-scale deep network. In *Proceedings of the 28th International Conference on Neural Information Processing Systems*; MIT Press: Cambridge, MA, USA, 2014.
9. Zhao, C.; Zhang, Y.; Poggi, M.; Tosi, F.; Guo, X.; Zhu, Z.; Huang, G.; Tang, Y.; Mattoccia, S. Monovit: Self-supervised monocular depth estimation with a vision transformer. In *Proceedings of the 2022 International Conference on 3D Vision (3DV)*; IEEE: Piscataway, NJ, USA, 2022; pp. 668–678. https://doi.org/10.1109/3dv57658.2022.00077.
10. Cheng, Z.; Zhang, Y.; Tang, C. Swin-depth: Using transformers and multi-scale fusion for monocular-based depth estimation. *IEEE Sensors J.* **2021**, *21*, 26912–26920. https://doi.org/10.1109/jsen.2021.3120753.
11. Duan, Y.; Guo, X.; Zhu, Z. Diffusiondepth: Diffusion denoising approach for monocular depth estimation. In *Proceedings of the European Conference on Computer Vision*; Springer: Cham, Switzerland, 2024; pp. 432–449. https://doi.org/10.1007/978-3-031-73247-8_25.
12. Li, P.; Ding, Y.; Wang, H.; Tang, C.; Li, Z. The devil is in the edges: Monocular depth estimation with edge-aware consistency fusion. *arXiv* **2024**, arXiv:2404.00373.
13. Saxena, S.; Kar, A.; Norouzi, M.; Fleet, D.J. Monocular depth estimation using diffusion models. *arXiv* **2023**, arXiv:2302.14816.
14. Ke, B.; Obukhov, A.; Huang, S.; Metzger, N.; Daudt, R.C.; Schindler, K. Repurposing diffusion-based image generators for monocular depth estimation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*; IEEE: Piscataway, NJ, USA, 2024; pp. 9492–9502. https://doi.org/10.1109/cvpr52733.2024.00907.
15. Skladan, M.; Chudá, J.; Singh, A.; Masný, M.; Lieskovský, M.; Pástor, M.; Mokroš, M.; Výboštok, J. Choosing the right close-range technology for measuring DBH in fast-growing trees plantations. *Trees For. People* **2025**, *19*, 100747. https://doi.org/10.1016/j.tfp.2024.100747.
16. Holvoet, J.; Eichhorn, M.P.; Giannetti, F.; Kükenbrink, D.; Liang, X.; Mokroš, M.; Novotný, J.; Pitkänen, T.P.; Puliti, S.; Skudnik, M.; et al. Terrestrial and mobile laser scanning for national forest inventories: From theory to implementation. *Remote Sens. Environ.* **2025**, *329*, 114947. https://doi.org/10.1016/j.rse.2025.114947.
17. Liang, X.; Hyyppä, J.; Kaartinen, H.; Lehtomäki, M.; Pyörälä, J.; Pfeifer, N.; Holopainen, M.; Brolly, G.; Francesco, P.; Hackenberg, J.; et al. International benchmarking of terrestrial laser scanning approaches for forest inventories. *ISPRS J. Photogramm. Remote Sens.* **2018**, *144*, 137–179. https://doi.org/10.1016/j.isprsjprs.2018.06.021.
18. Yang, S.; Xing, Y.; Yin, B.; Wang, D.; Chang, X.; Wang, J. A Novel Method for Extracting DBH and Crown Base Height in Forests Using Small Motion Clips. *Forests* **2024**, *15*, 1635. https://doi.org/10.3390/f15091635.
19. Grondin, V.; Fortin, J.M.; Pomerleau, F.; Giguère, P. Tree detection and diameter estimation based on deep learning. *Forestry* **2023**, *96*, 264–276. https://doi.org/10.1093/forestry/cpac043.
20. Hou, J.; Zhou, H.; Hu, J.; Yu, H.; Hu, H. A Multi-Scale Convolution and Multi-Layer Fusion Network for Remote Sensing Forest Tree Species Recognition. *Remote Sens.* **2023**, *15*, 4732. https://doi.org/10.3390/rs15194732.
21. Hou, J.; Zhou, H.; Yu, H.; Hu, H. HPAC: A Forest Tree Species Recognition Network Based on Multi-Scale Spatial Enhancement in Remote Sensing Images. *Int. J. Remote Sens.* **2023**, *44*, 5960–5975. https://doi.org/10.1080/01431161.2023.2257861.
22. Ye, Y.; Zhou, H.; Yu, H.; Hu, H.; Zhang, G.; Hu, J.; He, T. Application of Tswin-F Network Based on Multi-Scale Feature Fusion in Tomato Leaf Lesion Recognition. *Pattern Recognit.* **2024**, *156*, 110775. https://doi.org/10.1016/j.patcog.2024.110775.
23. Zhou, H.; Chen, C.; Xia, Z.; Ding, Q.; Liao, Q.; Wang, Q.; Yu, H.; Hu, H.; Zhang, G.; Hu, J.; et al. HGCS-Det: A Deep Learning-Based Solution for Localizing and Recognizing Household Garbage in Complex Scenarios. *Sensors* **2025**, *25*, 3726. https://doi.org/10.3390/s25123726.
24. Lin, L.; Ding, Q.; Li, X.; Hu, H.; Wang, Q.; Zhou, H. GMamba: A Lightweight Mamba Model for Garbage Classification. *Sustainability* **2026**, *18*, 5397. https://doi.org/10.3390/su18115397.

25. Xu, D.; Ricci, E.; Ouyang, W.; Wang, X.; Sebe, N. Structured Attention Guided Convolutional Neural Fields for Monocular Depth Estimation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*; IEEE: Piscataway, NJ, USA, 2018; pp. 3917–3925.
26. Ramamonjisoa, M.; Lepetit, V. Sharpnet: Fast and accurate recovery of occluding contours in monocular depth estimation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision Workshops*; IEEE: Piscataway, NJ, USA, 2019; pp. 2109–2118. https://doi.org/10.1109/iccvw.2019.00266.
27. Godard, C.; Mac Aodha, O.; Firman, M.; Brostow, G.J. Digging into self-supervised monocular depth estimation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*; IEEE: Piscataway, NJ, USA, 2019; pp. 3828–3838. https://doi.org/10.1109/iccv.2019.00393.
28. Zhang, N.; Nex, F.; Vosselman, G.; Kerle, N. Lite-mono: A lightweight cnn and transformer architecture for self-supervised monocular depth estimation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*; IEEE: Piscataway, NJ, USA, 2023; pp. 18537–18546. https://doi.org/10.1109/cvpr52729.2023.01778.
29. Wofk, D.; Ma, F.; Yang, T.J.; Karaman, S.; Sze, V. Fastdepth: Fast monocular depth estimation on embedded systems. In *Proceedings of the 2019 International Conference on Robotics and Automation (ICRA)*; IEEE: Piscataway, NJ, USA, 2019; pp. 6101–6108. https://doi.org/10.1109/icra.2019.8794182.
30. Li, Y.; Wei, X. MobileDepth: Monocular depth estimation based on lightweight vision transformer. *Appl. Artif. Intell.* **2024**, *38*, 2364159. https://doi.org/10.1080/08839514.2024.2364159.
31. Hu, J.; Fan, C.; Jiang, H.; Guo, X.; Gao, Y.; Lu, X.; Lam, T.L. Boosting lightweight depth estimation via knowledge distillation. In *Proceedings of the International Conference on Knowledge Science, Engineering and Management*; Springer: Cham, Switzerland, 2023; pp. 27–39. https://doi.org/10.1007/978-3-031-40283-8_3.
32. Ji, Y.; Chen, Z.; Xie, E.; Hong, L.; Liu, X.; Liu, Z.; Lu, T.; Li, Z.; Luo, P. Ddp: Diffusion model for dense visual prediction. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*; IEEE: Piscataway, NJ, USA, 2023; pp. 21741–21752. https://doi.org/10.1109/iccv51070.2023.01987.
33. Liu, Z.; Lin, Y.; Cao, Y.; Hu, H.; Wei, Y.; Zhang, Z.; Lin, S.; Guo, B. Swin transformer: Hierarchical vision transformer using shifted windows. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*; IEEE: Piscataway, NJ, USA, 2021; pp. 10012–10022. https://doi.org/10.1109/iccv48922.2021.00986.
34. Lin, T.Y.; Dollár, P.; Girshick, R.; He, K.; Hariharan, B.; Belongie, S. Feature pyramid networks for object detection. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*; IEEE: Piscataway, NJ, USA, 2017; pp. 2117–2125. https://doi.org/10.1109/cvpr.2017.106.
35. Sobel, I.; Feldman, G. A 3 × 3 isotropic gradient operator for image processing. *Talk Stanf. Artif. Proj.* **1968**, *1968*, 271–272.
36. Zhang, H.; Dun, Y.; Pei, Y.; Lai, S.; Liu, C.; Zhang, K.; Qian, X. HF-HRNet: A simple hardware friendly high-resolution network. *IEEE Trans. Circuits Syst. Video Technol.* **2024**, *34*, 7699–7711. https://doi.org/10.1109/tcsvt.2024.3377365.
37. Kirillov, A.; Mintun, E.; Ravi, N.; Mao, H.; Rolland, C.; Gustafson, L.; Xiao, T.; Whitehead, S.; Berg, A.C.; Lo, W.Y.; et al. Segment Anything. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*; IEEE: Piscataway, NJ, USA, 2023; pp. 4015–4026.
38. Zhang, Z. A flexible new technique for camera calibration. *IEEE Trans. Pattern Anal. Mach. Intell.* **2002**, *22*, 1330–1334. https://doi.org/10.1109/34.888718.
39. Geiger, A.; Lenz, P.; Stiller, C.; Urtasun, R. Vision meets robotics: The kitti dataset. *Int. J. Robot. Res.* **2013**, *32*, 1231–1237. https://doi.org/10.1177/0278364913491297.
40. Feng, Z.; She, Y.; Keshav, S. SPREAD: A large-scale, high-fidelity synthetic dataset for multiple forest vision tasks. *Ecol. Inform.* **2025**, *87*, 103085. https://doi.org/10.2139/ssrn.4977986.
41. Paszke, A.; Gross, S.; Massa, F.; Lerer, A.; Bradbury, J.; Chanan, G.; Killeen, T.; Lin, Z.; Gimelshein, N.; Antiga, L.; et al. Pytorch: An imperative style, high-performance deep learning library. In *Proceedings of the 33rd International Conference on Neural Information Processing Systems*; Curran Associates Inc.: Red Hook, NY, USA, 2019.
42. Wang, C.; Buenaposada, J.M.; Zhu, R.; Lucey, S. Learning depth from monocular videos using direct methods. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*; IEEE: Piscataway, NJ, USA, 2018; pp. 2022–2030. https://doi.org/10.1109/cvpr.2018.00216.
43. Yin, Z.; Shi, J. Geonet: Unsupervised learning of dense depth, optical flow and camera pose. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*; IEEE: Piscataway, NJ, USA, 2018; pp. 1983–1992. https://doi.org/10.1109/cvpr.2018.00212.
44. Lyu, X.; Liu, L.; Wang, M.; Kong, X.; Liu, L.; Liu, Y.; Chen, X.; Yuan, Y. Hr-depth: High resolution self-supervised monocular depth estimation. In *Proceedings of the AAAI Conference on Artificial Intelligence*; AAAI: Washington, DC, USA, 2021; Volume 35, pp. 2294–2301. https://doi.org/10.1609/aaai.v35i3.16329.

45. Yan, J.; Zhao, H.; Bu, P.; Jin, Y. Channel-wise attention-based network for self-supervised monocular depth estimation. In *Proceedings of the 2021 International Conference on 3D Vision (3DV)*; IEEE: Piscataway, NJ, USA, 2021; pp. 464–473. https://doi.org/10.1109/3dv53792.2021.00056.
46. Zhou, H.; Greenwood, D.; Taylor, S. Self-supervised monocular depth estimation with internal feature fusion. *arXiv* **2021**, arXiv:2110.09482. https://doi.org/10.5244/c.35.208.
47. Cheng, Z.; Zhang, Y.; Yu, Y.; Song, Z.; Tang, C. TinyDepth: Lightweight self-supervised monocular depth estimation based on transformer. *Eng. Appl. Artif. Intell.* **2024**, *138*, 109313. https://doi.org/10.1016/j.engappai.2024.109313.
48. He, M.; Hui, L.; Bian, Y.; Ren, J.; Xie, J.; Yang, J. Ra-depth: Resolution adaptive self-supervised monocular depth estimation. In *Proceedings of the European Conference on Computer Vision*; Springer: Cham, Switzerland, 2022; pp. 565–581. https://doi.org/10.1007/978-3-031-19812-0_33.
49. Tu, X.; Xu, C.; Liu, S.; Li, R.; Xie, G.; Huang, J.; Yang, L.T. Efficient monocular depth estimation for edge devices in internet of things. *IEEE Trans. Ind. Inform.* **2020**, *17*, 2821–2832. https://doi.org/10.1109/tii.2020.3020583.
50. Rudolph, M.; Dawoud, Y.; Güldenring, R.; Nalpantidis, L.; Belagiannis, V. Lightweight monocular depth estimation through guided decoding. In *Proceedings of the 2022 International Conference on Robotics and Automation (ICRA)*; IEEE: Piscataway, NJ, USA, 2022; pp. 2344–2350. https://doi.org/10.1109/icra46639.2022.9812220.
51. Lee, D.J.; Lee, J.Y.; Shon, H.; Yi, E.; Park, Y.H.; Cho, S.S.; Kim, J. Lightweight monocular depth estimation via token-sharing transformer. *arXiv* **2023**, arXiv:2306.05682. https://doi.org/10.1109/icra48891.2023.10160566.
52. Zhou, Z.; Fan, X.; Shi, P.; Xin, Y. R-msfm: Recurrent multi-scale feature modulation for monocular depth estimating. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*; IEEE: Piscataway, NJ, USA, 2021; pp. 12777–12786. https://doi.org/10.1109/iccv48922.2021.01254.
53. Klingner, M.; Termöhlen, J.A.; Mikolajczyk, J.; Fingscheidt, T. Self-supervised monocular depth estimation: Solving the dynamic object problem by semantic guidance. In *Proceedings of the European Conference on Computer Vision*; Springer: Cham, Switzerland, 2020; pp. 582–600. https://doi.org/10.1007/978-3-030-58565-5_35.
54. Li, Z.; Chen, Z.; Li, A.; Fang, L.; Jiang, Q.; Liu, X.; Jiang, J.; Zhou, B.; Zhao, H. Simipu: Simple 2D image and 3D point cloud unsupervised pre-training for spatial-aware visual representations. In *Proceedings of the AAAI Conference on Artificial Intelligence*; AAAI: Washington, DC, USA, 2022; Volume 36, pp. 1500–1508.
55. Yin, W.; Liu, Y.; Shen, C.; Yan, Y. Enforcing geometric constraints of virtual normal for depth prediction. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*; IEEE: Piscataway, NJ, USA, 2019; pp. 5684–5693. https://doi.org/10.1109/iccv.2019.00578.
56. Lee, J.H.; Han, M.K.; Ko, D.W.; Suh, I.H. From big to small: Multi-scale local planar guidance for monocular depth estimation. *arXiv* **2019**, arXiv:1907.10326.
57. Bhat, S.F.; Alhashim, I.; Wonka, P. Adabins: Depth estimation using adaptive bins. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*; IEEE: Piscataway, NJ, USA, 2021; pp. 4009–4018. https://doi.org/10.1109/cvpr46437.2021.00400.
58. Ranftl, R.; Bochkovskiy, A.; Koltun, V. Vision transformers for dense prediction. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*; IEEE: Piscataway, NJ, USA, 2021; pp. 12179–12188. https://doi.org/10.1109/iccv48922.2021.01196.
59. Agarwal, A.; Arora, C. Depthformer: Multiscale vision transformer for monocular depth estimation with global local information fusion. In *Proceedings of the 2022 IEEE International Conference on Image Processing (ICIP)*; IEEE: Piscataway, NJ, USA, 2022; pp. 3873–3877. https://doi.org/10.1109/icip46576.2022.9897187.
60. Thompson, J.L.; Phung, S.L.; Bouzerdoum, A. D-net: A generalised and optimised deep network for monocular depth estimation. *IEEE Access* **2021**, *9*, 134543–134555. https://doi.org/10.1109/access.2021.3116380.
61. Li, Z.; Chen, Z.; Liu, X.; Jiang, J. Depthformer: Exploiting long-range correlation and local information for accurate monocular depth estimation. *Mach. Intell. Res.* **2023**, *20*, 837–854. https://doi.org/10.1007/s11633-023-1458-0.
62. She, Y.; Li, P.; Wei, M.; Liang, D.; Chen, Y.; Xie, H.; Wang, F.L. eViTBins: Edge-Enhanced Vision-Transformer Bins for Monocular Depth Estimation on Edge Devices. *IEEE Trans. Intell. Transp. Syst.* **2024**, *25*, 20320–20334. https://doi.org/10.1109/tits.2024.3480114.
63. Zhang, C.; Han, D.; Qiao, Y.; Kim, J.U.; Bae, S.H.; Lee, S.; Hong, C.S. Faster Segment Anything: Towards Lightweight SAM for Mobile Applications. *arXiv* **2023**, arXiv:2306.14289. https://doi.org/10.48550/arXiv.2306.14289.