IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received 5 March 2024, accepted 24 March 2024, date of publication 27 March 2024, date of current version 3 April 2024.

Digital Object Identifier 10.1109/ACCESS.2024.3382572

RESEARCH ARTICLE

Multi-Scale Monocular Depth Estimation Based on Global Understanding

JIEJIE XIAO[1], LIHONG LI[2], XU SU[1], AND GUOPENG TAN[1]
[1]School of Information and Electrical Engineering, Hebei University of Engineering, Handan 056038, China
[2]Hebei Key Laboratory of Security and Protection Information Sensing and Processing, Hebei University of Engineering, Handan 056038, China
Corresponding author: Lihong Li (llhxinxi@126.com)

ABSTRACT With the advancement of Convolutional Neural Networks, numerous convolutional neural network-based methods have been proposed for depth estimation and have achieved significant achievements. However, the repetitive convolutional layers and spatial pooling layers in these networks often lead to a reduction in spatial resolution and loss of local information, such as edge contours. To address this issue, this study presents a multi-scale monocular depth estimation model. Specifically, a Global Understanding Module was introduced on top of a generic encoder to increase the receptive field and capture contextual information. Additionally, the decoding process incorporates a Difference Module and a Multi-scale Cascade Module to guide the decoding information and refine edge contour details. Finally, extensive experiments were conducted using the KITTI and NYUv2 datasets. For the KITTI dataset, the Absolute Relative Error (Abs. Rel) was 0.057, and the Root Mean Squared Error (RMSE) was 2.415. On the NYUv2 dataset, Abs.Rel was 0.104, and RMSE was 0.380. These results indicate that the model performs well in accurately estimating depth information.

INDEX TERMS Convolutional neural networks, depth estimation, global understanding module, difference module, cascade module.

I. INTRODUCTION

Monocular depth estimation aims to infer the depth map of a scene using only a single RGB image. However, monocular depth estimation is considered an ill-posed problem because a single 2D image can correspond to multiple 3D scenes. Consequently, compared to depth estimation from stereo images, the progress in monocular depth estimation has been relatively slow. In the early stages, traditional methods utilized monocular cues for depth prediction, such as viewpoint and texture information [1], [2], or selected appropriate depth values by leveraging similarity with other scene structures. However, these methods could not still accurately predict depth solely from a single image.

With the advancement of deep learning, deep-learning-based monocular depth estimation has gained increasing attention. To improve the performance of monocular depth estimation, researchers have adopted models based on deep neural networks (DNNs) [3], [4], [5] and have demonstrated the superiority of deep features over handcrafted features. DNN models applied fully convolutional architectures as feature extractors [5], [6]. However, repeated spatial pooling layers rapidly reduce the spatial resolution of feature maps, which is not ideal for depth estimation tasks. Although high-resolution depth maps can be obtained using multi-layer deconvolution networks [7] or skip connections [8], these approaches require additional computations and make the network architecture and training process more complex. To extract latent features related to depth information, Convolutional Neural Networks (CNNs) have been widely employed as the backbone structure of depth models. CNN-based depth networks typically consist of two parts: an encoder for feature extraction and a decoder for depth prediction. Commonly used encoders in the encoder-decoder network architecture include ResNet [9], DenseNet [10], and ResNeXt [11], which are used to extract latent features. These features are then simply upsampled to their original size and transformed into a depth map through the decoding process. However, this simple upsampling process fails to adequately consider the depth boundaries of objects at various scales, leading to blurriness of in-depth information at object boundaries.

The associate editor coordinating the review of this manuscript and approving it for publication was Tai-Hoon Kim.

To address these issues, this study proposes a multi-scale network architecture with global understanding. First, a backbone network is used to extract the encoding features from the RGB image. Then, a Global Understanding Module (GUM) is introduced at the high spatial resolution layers to fully utilize the encoding features. Finally, multiple cascade modules are employed to combine differencing information and decode information from different layers to estimate the depth of information. By aggregating global and local information effectively, the entire network structure can better handle depth boundaries and details, thereby improving the performance of monocular depth estimation.

The main contributions of this paper are as follows:

1.We propose a multi-scale global understanding network for monocular depth estimation that effectively leverages global and local information.

2.Introducing the global-understanding module expands the receptive field and captures contextual information. The combination of the Difference Module and Cascade Module captures the depth boundary contour information, further enhancing the accuracy of the depth estimation.

3.Extensive experiments were conducted using the KITTI and NYUv2 datasets to validate the effectiveness of the proposed model.

II. RELATED WORK

A. MONOCULAR DEPTH ESTIMATION

Depth estimation plays a crucial role in understanding the 3D information from RGB images. In the early stages, geometric algorithms [12], [13] were primarily used for depth estimation from stereo images, but these methods relied heavily on point correspondences between images. Subsequently, Saxena et al. [1] introduced Markov Random Fields (MRF) to model the relationships between the depth values of different pixels, pioneering the use of monocular cues for depth prediction. Researchers have proposed various handcrafted algorithms for monocular cue estimation [14], [15]. For example, Karsch et al. [14] utilized spectral coefficient similarity to determine candidate depth values, and Herrera et al. [15] employed a clustering-based learning approach to determine the optimal depth. However, in complex scenes, handcrafted features often fail to represent geometric structures clearly, resulting in blurred depth map estimates. With the advancement of deep learning, scholars have proposed numerous monocular depth estimation methods based on Convolutional Neural Networks (CNN) [16], [17]. Compared with other approaches, CNN-based monocular depth estimation networks demonstrate superior performance and ease of operation.

B. MONOCULAR DEPTH ESTIMATION BASED ON MULTI-SCALE

Initially, Eigen et al. [5], [18] designed a multi-scale network that improved the accuracy of monocular depth estimation to a new level by progressively refining the spatial resolution of the depth map. This laid the foundation for subsequent research, and an increasing number of scholars have begun developing various multi-scale network architectures to predict depth from a single image. For instance, Xu et al. [19] fused the information output of a multi-scale convolutional neural network with multiple Conditional Random Fields (CRF) in a cascaded manner to obtain more accurate depth information. To reduce the computational complexity of fully connected CRFs, NeW CRFs [20] utilized neural-window CRFs to refine depth information at different scales. Lee et al. [21] proposed a Local Plane Guided Layer that directly and explicitly guided the relationship between multi-scale features and the depth map, thereby improving the accuracy of depth estimation. Song et al. [22] employed the Laplacian pyramid decoding technique to estimate a clearer depth map by leveraging the feature information at different scales.

C. MONOCULAR DEPTH ESTIMATION BASED ON LOCAL AND GLOBAL INFORMATION

It is crucial for monocular depth estimation networks to effectively balance global and local information, which has been addressed by researchers through various methods to enhance the accuracy and detail recovery ability of depth estimation. Fu et al. [23] introduced the spacing-increasing discretization (SID) strategy for depth discretization and utilized the Atrous Spatial Pyramid Pooling (ASPP) scheme [24] to extract dense features. DiffusionDepth [25] employed hierarchical aggregation and heterogeneous interaction to enhance the feature information across scales. Lee et al. [26] proposed a Partitioned Attention Module to fuse spatial and channel information for improved depth-detail representation. Some researchers have introduced transformers to leverage global information effectively. For example, DPT [27] and PixelFormer [28] significantly improved performance with the introduction of transformers. Yang et al. [29] used transformers to extract global features and employed an attention-gate decoder to capture detailed information. In addition, various constraints are introduced to improve the algorithms. Patil et al. [30] proposed a method for selectively utilizing coplanar pixel information to enhance depth estimation. VA Depth [31] introduced variational inference for depth prediction. Bhat et al. [32] introduced uncertainty calibration and cross-distillation between transformers and convolutional neural networks to make full use of local and global information. Patil et al. [33] and Li et al. [34] refined local information by computing adaptive boxes. Depthformers [35] integrated the strengths of Transformers and CNNs to predict depth information by leveraging long-range correlations and local information.

In the field of monocular depth estimation, two primary approaches, Convolutional Neural Networks and transformers, are commonly utilized. However, both methods have their limitations. CNNs struggle to capture long-range information

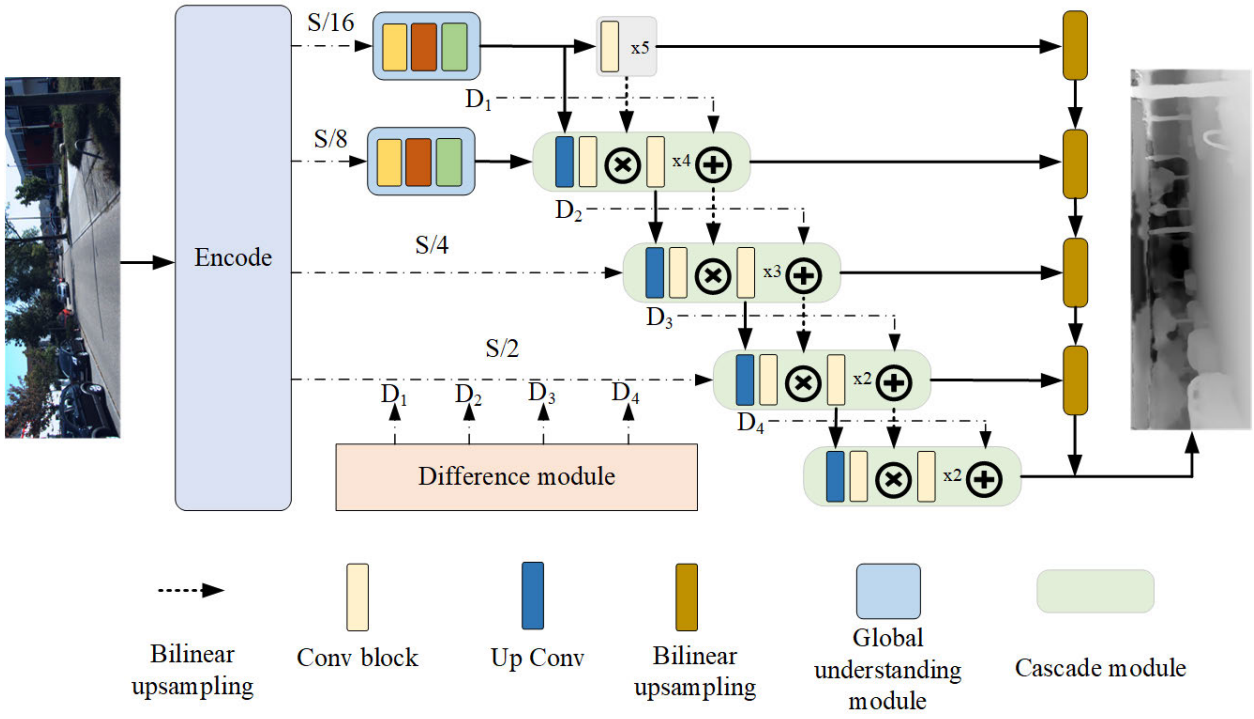


FIGURE 1. Block diagram of the network.

effectively, while transformers excel at capturing global information but may lose fine-grained details. Furthermore, transformers require substantial computational resources due to their powerful attention mechanisms. To address these issues, this study adopts a CNN as the backbone network and introduces a Global Understanding Module to expand the receptive field and fully consider global information. Simultaneously, in the cascaded module, details from the different modules are combined with features at different scales. This multi-scale network structure allows for the simultaneous consideration of global and local features, thereby restoring high-resolution details in the depth map.

III. PROPOSED METHOD

This section introduces a multi-scale cascaded network architecture designed to incorporate information at different scales, from global to local. The architecture comprises several key components. Firstly, an encoder is employed to extract dense features from the input. Secondly, a Global Understanding Module, as described in Section III-C, is incorporated to capture global contextual information effectively. This module enhances the network's ability to understand the overall scene. Thirdly, a Difference Module, outlined in Section III-D, is introduced to facilitate accurate estimation of boundary contours. Lastly, a Cascade Module, explained in Section III-B, is utilized to combine information from multiple scales, enabling comprehensive depth estimation.

A. NETWORK ARCHITECTURE

The network architecture is depicted in Figure 1. The encoder section utilizes a pre-trained model called ResNext101, which is a variant of ResNet. ResNext101 employs convolutional operations by dividing the original convolutional layers into distinct branches and merging them. After passing the RGB image through the encoder, dense features with four different spatial resolutions (S/2, S/4, S/8, and S/16) were obtained. To capture more contextual information, the two highest decoding layers are connected to the global understanding module. Finally, the Cascaded Module combines the four difference features extracted by the Difference Module with the corresponding decoding information to enhance the accuracy of depth estimation.

B. CASCADE MODULE

The Cascade Module achieves its functionality through a combination of convolutional blocks and is primarily used for aggregating the difference and decoding features. The specific operations are as follows: The Cascaded Module performs an upsampling operation on the information from the previous layer to match its size with the encoded features of the current scale. The upsampled features are then element-wise multiplied by the current scale's encoded features and depth features obtained through convolutional operations. Next, the difference information was added to the previous multiplication result to obtain the depth features at the current scale. This operation effectively integrates the detailed feature information from different

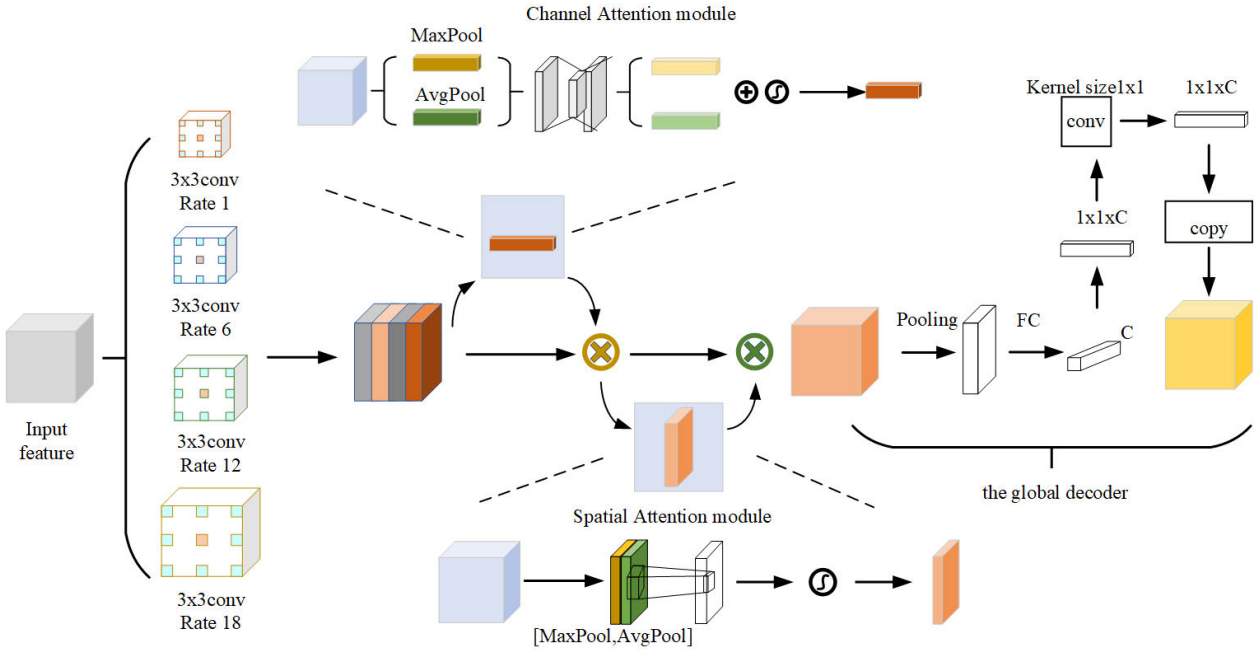


FIGURE 2. Global understanding module (GUM).

scales. Unlike traditional convolutional layers (convolution-activation-normalization), the convolutional layer in the Cascaded Module consists of the following steps: Layer Normalization, GELU activation function, and 7×7 convolution. It is important to note that the number of convolutions varied for each decoding layer. For example, the fourth decoding layer consists of five convolutional layers, the third decoding layer consists of four convolutional layers, etc. Each convolutional layer fuses the detailed feature information from different scales, resulting in more accurate depth features.

C. GLOBAL UNDERSTANDING MODULE

The Global Understanding Module consists of three components: Atrous Spatial Pyramid Pooling (ASPP), a Spatial Channel Learner, and a Global Decoder, as shown in Figure 2. The first is the ASPP, which consists of atrous convolutions with dilation rates of 1, 6, 12, and 18, followed by the BN layers and ReLU activation functions. ASPP captures features at different receptive fields and provides multi-scale contextual information by using different dilation rates. Next is the Spatial Channel Learner, which aims to add attention mechanisms on both the channel and spatial dimensions to facilitate the fusion of complex feature information. The Spatial Channel Learner first performs max-pooling and average-pooling operations in the channel dimension, resulting in two 1×1×C feature maps. These two feature maps are then added element-wise through a two-layer weight-sharing network (convolution-ReLU-convolution) and passed through a sigmoid activation function to generate channel feature information. Subsequently, the channel feature information is assigned spatial weights in the spatial dimension, resulting in the final channel-spatial feature. Finally, the Global Decoder reduces the spatial dimension using an average pooling layer and obtains a C-channel feature vector through fully connected layers. This feature vector was then used as a pooling structure by a 1×1 convolutional layer, and the feature information was replicated along the spatial dimension to achieve a comprehensive understanding of the input image.

D. DIFFERENCE MODULE

The main purpose of the Difference Module is to predict local details and boundary contour information. The specific steps are as follows. First, the RGB image is inputted and undergoes a series of downsampling operations to obtain feature information with spatial resolutions of 1/2, 1/4, 1/8, and 1/16 of the input image resolution. Next, through a progressive upsampling process, the feature maps were restored to the original resolution of the input image, ensuring scale consistency. Finally, a subtraction operation was performed between the feature maps of the same resolution to obtain the desired local boundary contour information. This operation highlights the boundaries and detailed regions in the image, providing richer depth estimation information. The structure of the model is shown in Figure 3.

E. LOSS FUNCTION

The experiment utilizes the scale-invariant log scale loss [18] to optimize the model and is obtained by calculating the difference between the predicted depth value y_i in the log

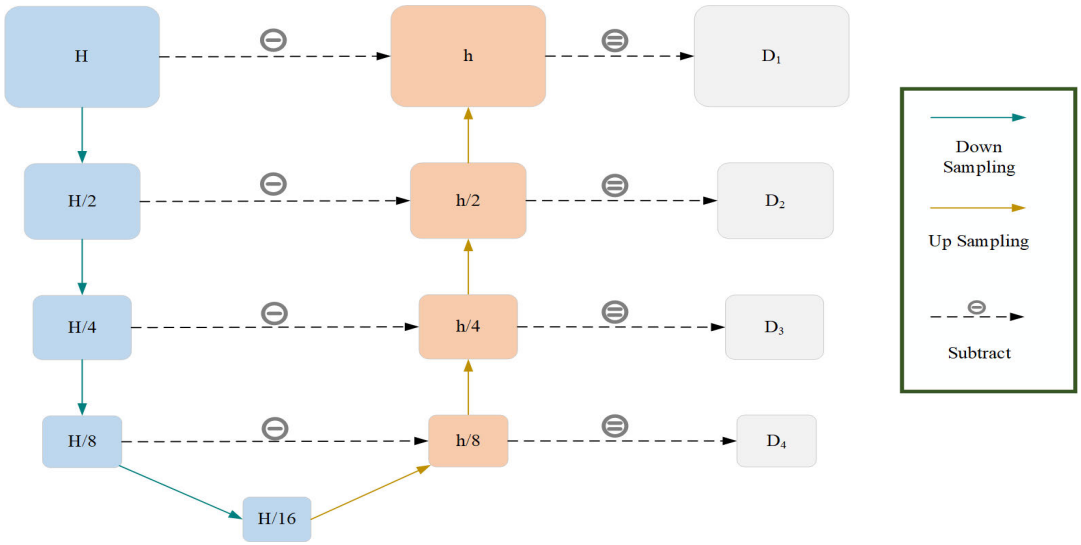


FIGURE 3. Difference module.

space and the ground truth y_i':

$$D(y_i, y_i') = \frac{1}{N}\sum_i d_i^2 - \frac{1}{N^2}\left(\sum_i d_i\right)^2. \tag{1}$$

The above equation can be written as the sum of the variance of the error and the weighted mean in logarithmic space:

$$D(y_i, y_i') = \frac{1}{N}\sum_i d_i^2 - \left(\frac{1}{N}\sum_i d_i\right)^2 + (1-\lambda)\left(\frac{1}{N}\sum_i d_i\right)^2. \tag{2}$$

where $d_i = log_{y_i} - log_{y_i'}$, N denotes the number of pixels with valid ground truth.

To focus more on minimizing the variance of the error, a higher λ can be set; therefore, λ is set to 0.85. Finally, the loss function is set to L, and is defined by the following formula:

$$L = \alpha\sqrt{D(y_i, y_i')}. \tag{3}$$

where α is constant and set to 10.

IV. EXPERIMENT

In this section, the performance of the overall network is validated and evaluated using two datasets: KITTI [36]and NYU Depth V2 [37]. The KITTI dataset consists of large-scale outdoor scenes with 61 different categories. Among these, 697 images covering 29 scenes were used for evaluation, whereas the remaining 23,488 images from 32 scenes were used for training. The NYU Depth V2 dataset, on the other hand, comprises indoor scenes. It included a total of 464 scenes, with 249 scenes used for training and the remaining 215 scenes used for testing. This dataset contains 654 images from these scenes. The specific training and testing procedures follow the partitioning approach proposed in [18].

A. EXPERIMENTAL DETAILS

The model was trained and tested in Python 3.8, PyTorch version 1.12.0, using CUDA 11.3. Experiments with the KITTI dataset were conducted on four NVIDIA GeForce 3060-12 GPU devices, with a batch size of 8. The segmentation strategy introduced by Eigen et al. [18], based on the KITTI dataset, was used to evaluate the model, with K set to 80. For the experiments with the NYUv2 dataset, 4 Tesla V100 GPU devices were used, with a batch size of 16. The Adam optimizer [38] was employed, with a power of 0.9, momentum of 0.999, and an initial learning rate of 10^{-4}, which decayed to 10^{-5} at the end. The model was trained for 25 epochs on the KITTI dataset and 35 epochs on the NYUv2 dataset, including 5 pre-training epochs.

B. QUANTITATIVE ASSESSMENT

To evaluate the model effectively, this study adopted six performance metrics introduced in a previous study [18]. These metrics have been widely used in the in-depth estimation evaluations of the NYUv2 dataset. For the KITTI dataset, two standard metrics were required: Squared Relative Difference, and Root Mean Square Logarithmic Error. These metrics are defined in equations (4)-(9).

Threshold accuracy (δ), where $thr = 1.25, 1.25^2, 1.25^3$; d_i is the estimated value, and $\tilde{d}_i$ is the true value. The percentage of pixels in all d_i that are smaller than the threshold thr in the total pixels was counted. The closer the value is to 1, the better the effect.

The Root Mean Square Error ($RMSE$) is a conventional measure used to quantify regression errors. The $RMSE$log,

TABLE 1. Quantitative comparison of different models on the KITTI dataset.

Models	Abs.Rel	Sq.Rel	RMSE	RMSElog	$\delta_1 < 1.25$	$\delta_2 < 1.25^2$	$\delta_3 < 1.25^3$
	Lower is better				Higher is better		
Fu et al.[23]	0.072	0.307	2.727	0.120	0.932	0.984	0.994
BTS[21]	0.059	0.245	2.756	0.096	0.956	0.993	0.998
Lapdepth[22]	0.059	0.213	2.453	0.090	0.961	0.994	**0.999**
AdaBins[33]	0.067	0.278	2.960	0.103	0.949	0.992	0.998
DPT[27]	0.062	-	2.573	0.092	0.959	**0.995**	**0.999**
PWA[26]	0.060	0.221	2.604	0.093	0.958	0.994	**0.999**
P3depth[30]	0.071	0.270	2.842	0.103	0.953	0.993	0.998
Ours	**0.057**	**0.201**	**2.415**	**0.088**	**0.964**	**0.995**	**0.999**

TABLE 2. Quantitative comparison of different models on the NYUv2 dataset.

Models	Abs.Rel	Log10	RMSE	$\delta_1 < 1.25$	$\delta_2 < 1.25^2$	$\delta_3 < 1.25^3$
	Lower is better			Higher is better		
Fu et al.[23]	0.115	0.047	0.352	0.828	0.965	0.992
BTS[21]	0.112	0.047	0.393	0.882	0.979	0.995
Lapdepth[22]	0.110	0.047	0.393	0.885	0.979	0.995
PWA[26]	0.105	**0.045**	0.374	0.892	0.981	0.995
P3depth[30]	**0.104**	0.049	**0.356**	0.890	0.981	**0.996**
Ours	**0.104**	**0.045**	0.380	**0.895**	**0.983**	**0.996**

TABLE 3. Quantitative comparison of the Global Understanding module on the KITTI dataset using different backbone encoders.

Models	Abs.Rel	Sq.Rel	RMSE	RMSElog	$\delta_1 < 1.25$	$\delta_2 < 1.25^2$	$\delta_3 < 1.25^3$
	Lower is better				Higher is better		
ResNet101	0.063	0.225	2.494	0.094	0.957	0.993	0.999
ResNet101+GUM	0.060	0.211	2.468	0.092	0.960	0.994	0.999
DenseNet161	0.060	0.216	2.476	0.092	0.957	0.994	0.999
DenseNet161+GUM	0.060	0.211	2.435	0.091	0.961	0.994	0.999
MobileNetV2	0.072	0.284	2.760	0.109	0.941	0.990	0.998
MobileNetV2+GUM	0.068	0.249	2.631	0.103	0.949	0.992	0.998

on the other hand, introduces a relative aspect to this error, reducing the influence of large errors concerning distance.

Relative error (*Sq.Rel*) penalizes larger depth errors by squaring them. Absolute relative error (*Abs.Rel*): this calculates the normalized per-pixel error based on the ground truth depth, reducing the influence of large errors concerning distance.

The Log Error represents the average absolute value of the logarithmic difference between the predicted depth and the true depth, often denoted as log 10.

$$Threshold = \%of\,\tilde{d}_i, s.t.max\left(\frac{\tilde{d}_i}{d_i}, \frac{d_i}{\tilde{d}_i}\right) = \delta < thr. \quad (4)$$

$$RMSE = \sqrt{\frac{1}{|T|}\sum_{\hat{d}\in I}\frac{\left\|\tilde{d}-d\right\|^2}{d}}. \quad (5)$$

$$RMSE\log = \sqrt{\frac{1}{|T|}\sum_{\tilde{d}\in T}||\log\tilde{d}-\log d||^2}. \quad (6)$$

$$Sq.Rel = \frac{1}{|T|}\sum_{\hat{d}\in T}\frac{\|\tilde{d}-d\|^2}{d}. \quad (7)$$

$$Abs.Rel = \frac{1}{|T|}\sum_{\hat{d}\in T}\frac{\left|\tilde{d}-d\right|}{d}. \quad (8)$$

$$\log 10 = \frac{1}{|T|}\sum_{\tilde{d}\in T}|\log_{10}\tilde{d}-\log_{10}d\,|. \quad (9)$$

The algorithm proposed in this study was quantitatively evaluated against several classical algorithms, and the results are presented in Table 1. On the KITTI dataset, compared to Lapdepth [24], the model in this study achieved an improvement of 0.003 in accuracy $\delta_1 < 1.25$ and a decrease of 0.038 in RMSE, indicating better prediction accuracy for correct pixels. Additionally, the average relative error decreased by 0.012, suggesting that the network exhibited more stable predictions for depth variations.

On the NYUv2 dataset, compared to Lapdepth [24], the model in this study achieved an improvement of 0.014 in accuracy $\delta_1 < 1.25$ and an improvement of 0.005 at higher accuracy $\delta_2 < 1.25^2$ thresholds. The RMSE also decreases by 0.013, as shown in Table 2. The effectiveness and robustness of the model presented in this paper were demonstrated by evaluating the proposed method on these two benchmark datasets.

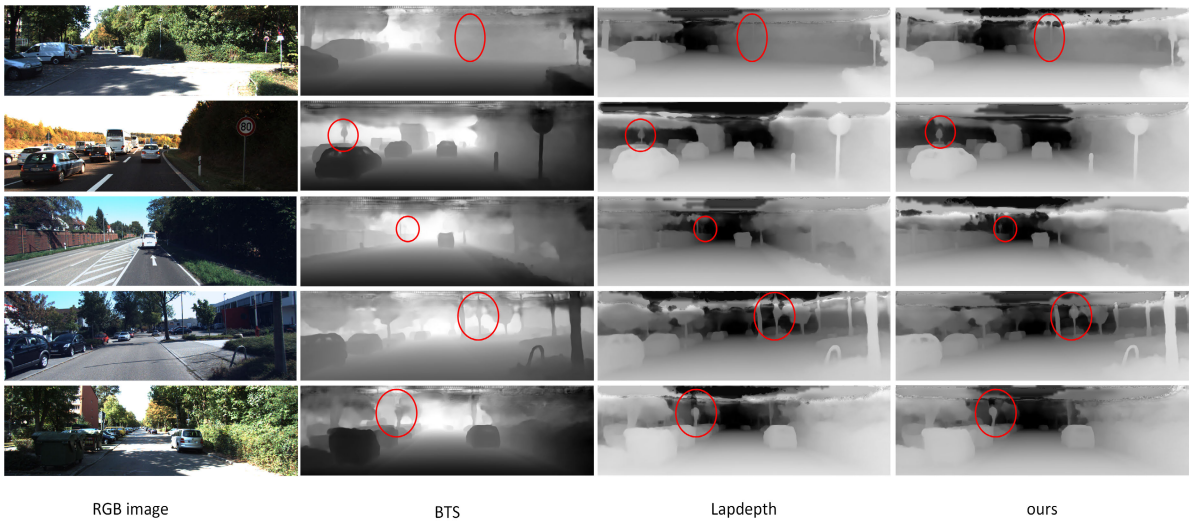


FIGURE 4. Qualitative comparison of depth estimation on the KITTI dataset. 1st colum: input color images, 2nd colum: results by BTS, 3rd colum: results by Lapdepth, 4th colum: results by proposed model.

TABLE 4. Quantity Verification of Global Understanding Modules on the NYUv2 Dataset.

Models	Abs.Rel	Log10	RMSE	$\delta_1 < 1.25$	$\delta_2 < 1.25^2$	$\delta_3 < 1.25^3$
	Lower is better			Higher is better		
GUM=0	0.108	0.047	0.388	0.891	0.982	0.995
GUM=1	0.105	0.045	0.381	0.893	0.984	0.996
GUM=2	0.104	0.045	0.380	0.895	0.984	0.996
GUM=3	0.105	0.045	0.384	0.893	0.983	0.996

TABLE 5. Quantitative comparison of the Global Understanding module using different backbone networks on the NYUv2 dataset.

Models	Abs.Rel	Log10	RMSE	$\delta_1 < 1.25$	$\delta_2 < 1.25^2$	$\delta_3 < 1.25^3$
	Lower is better			Higher is better		
BTS[21]	0.112	0.047	0.393	0.882	0.979	0.995
BTS+GUM	0.109	0.043	0.386	0.885	0.981	0.995
NEWCRFs[20]	0.095	0.041	0.338	0.921	0.990	0.998
NEWCRFs+GUM	0.092	0.038	0.331	0.922	0.991	0.998

C. QUALITATIVE ASSESSMENT

The qualitative comparison results for the KITTI dataset are shown in Figure 4. It is evident that the depth maps estimated by the proposed method not only predict distant objects (e.g., the pole in the middle of the first row and the sign at a far distance in the third row) but also capture detailed information about nearby objects (e.g., the sign in the last row). In addition, the overall depth boundaries obtained are relatively smooth. These results demonstrate that the proposed method can provide accurate depth information for depth estimations.

Qualitative comparison results for the NYUv2 dataset are shown in Figure 5. It can be clearly seen that the proposed method outperforms the Lapdepth method in terms of visualized results and is closer to the ground truth. The proposed method achieves more accurate predictions of object boundaries and details, such as hanging objects on the wall in the second row and objects on the table in the fourth row. These results further confirm the superiority of the network for depth estimation in indoor scenes.

D. ASSESSMENT GLOBAL UNDERSTANDING MODULE

To validate the generalization and effectiveness of the Global Understanding Module, different backbone networks were chosen as feature extractors, including ResNet [9], DenseNet [10], and MobileNetV2 [39]. Table 3 presents the quantitative results using different encoders, where the 3rd, 5th, and 7th rows show the performance evaluation results of networks using ResNet101, DenseNet161, and MobileNetV2 as encoders, and the 4th, 6th, and 8th rows show the results after introducing the Global Understanding Module to these encoders. From Table 3, it can be observed that the performance of the model improved after introducing the GUM, regardless of the encoder used. This demonstrates the effectiveness of the GUM in the task of depth estimation and showcases its generalization ability across different encoders.

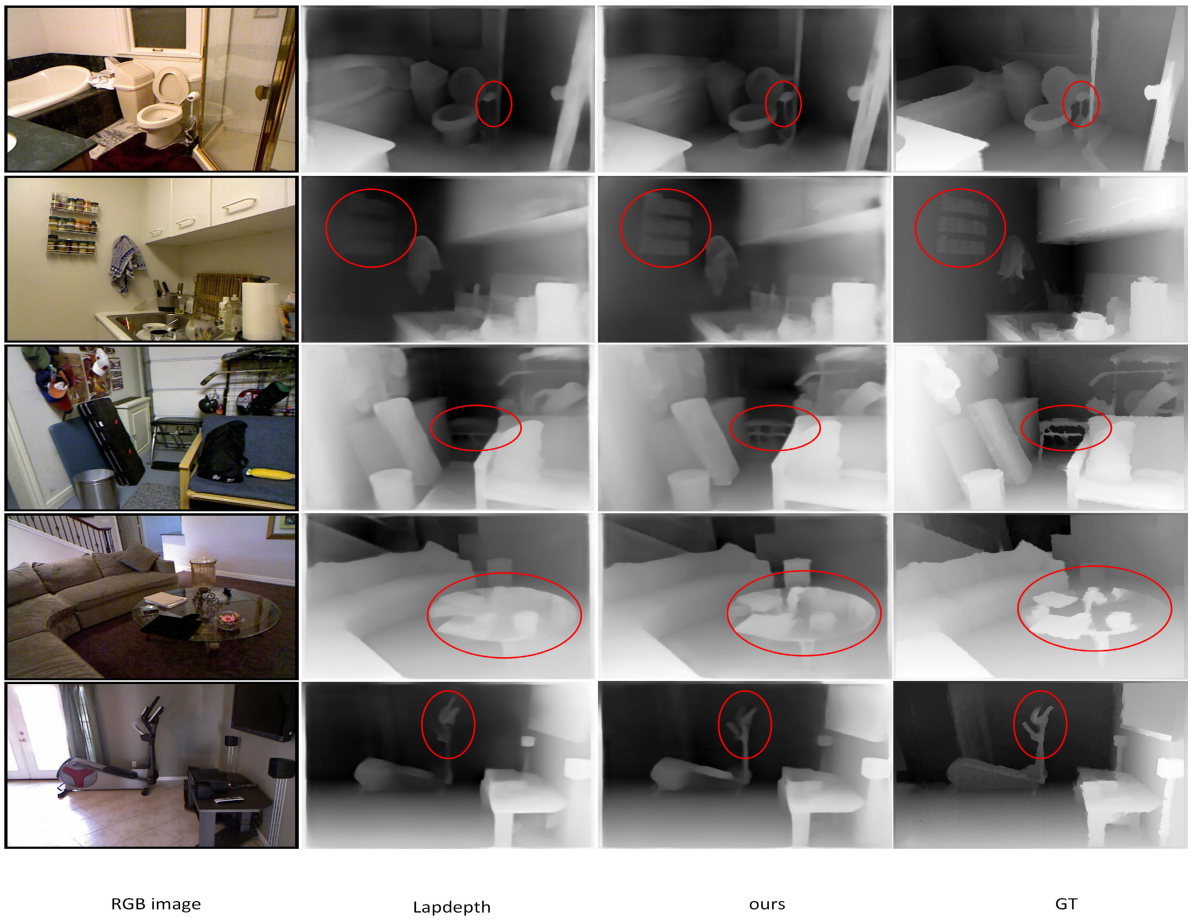


FIGURE 5. Qualitative comparison of depth estimation on the NYUv2 dataset. 1st colum: input color images, 2nd colum: results by Lapdepth, 3rd colum: results by proposed model, 4th colum: Ground truth.

Furthermore, this study evaluated the Global Understanding Modules in terms of their quantity. Using the Multi-scale Cascade network as the baseline, the performance was quantitatively assessed by progressively adding GUM modules at the decoding layers, as presented in Table 4. The results demonstrate a notable performance improvement when two GUM modules are introduced, indicating that GUM is highly effective in capturing high-level feature information.

Finally, this study applies the Global Understanding Module to two classical network models: the BTS model with CNN as the backbone network and the NeW CRFs model with Transform as the backbone network. The results are shown in Table 5, indicating that the performance of both models improved after introducing GUM, regardless of the backbone structure. In summary, the effectiveness and reliability of the GUM were further validated through three different experiments.

E. ABLATION EXPERIMENT OF CONVOLUTION

Because the cascade module is composed of convolutional blocks, this study compared different convolutional layers for more effective feature decoding, as shown in Figure 6. (a), (b), and (c) in the figure represent the convolutional layers with different structures. (a) consists of BN normalization, ReLU activation function, and 3×3 convolution; (b) consists of LN normalization, GELU activation function, and 3×3 convolution; (c) consists of LN normalization, GELU activation function, and 7×7 convolution. The data in Table 6 were obtained by comparing the quantitative results of the different convolutional layers. The table shows that the 7×7 convolutional layer with the GELU activation function and LN normalization can guide the decoding information more effectively.

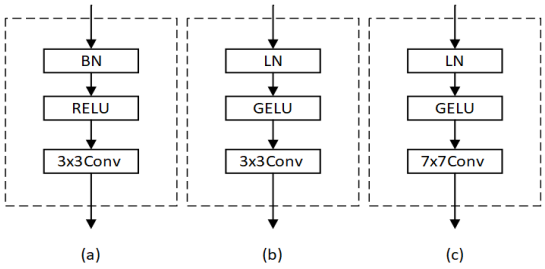


FIGURE 6. Convolution layers of different structures.

TABLE 6. The concatenation module uses different convolution blocks for quantitative comparison on the NYUv2 dataset.

Models	Abs.Rel	Log10	RMSE	$\delta_1 < 1.25$	$\delta_2 < 1.25^2$	$\delta_3 < 1.25^3$
		Lower is better			Higher is better	
(a)	0.106	0.045	0.384	0.891	0.983	0.995
(b)	0.106	0.046	0.385	0.893	0.983	0.996
(c)	0.104	0.045	0.380	0.895	0.984	0.996

TABLE 7. Quantitative comparison of different downsampling methods on NYUv2 dataset.

Models	Abs.Rel	Log10	RMSE	$\delta_1 < 1.25$	$\delta_2 < 1.25^2$	$\delta_3 < 1.25^3$
		Lower is better			Higher is better	
bicubic	0.107	0.046	0.386	0.890	0.983	0.996
bilinear	0.104	0.045	0.380	0.895	0.984	0.996

In addition, the difference module contains a large number of downsampling operations. To investigate the impact of different downsampling methods on depth information, this study compares bilinear and bicubic downsampling methods. Table 7 presents the quantitative results, clearly indicating that the bilinear downsampling method is more suitable for depth-prediction tasks than the bicubic downsampling method. Theoretically, the bicubic downsampling method can better aggregate information around pixels, whereas the interpolation effect of the bilinear downsampling method may be relatively poor. However, in the network structure proposed in this paper, with multi-scale decoding layers, using the bicubic downsampling method introduces more erroneous pixel information, ultimately leading to larger depth estimation errors in the lower layers.

V. CONCLUSION

In this paper, an effective multi-scale monocular depth estimation network is proposed that utilizes the Global Understanding Module and cascade module to leverage both long-range feature information and local information and recover the depth map without reducing the resolution. Through various comparative and evaluation experiments, the effectiveness of the GUM and reliability of the network were demonstrated. Additionally, visual analysis was performed, and it was observed that the estimated object boundaries were smoother and clearer, and that the network performed well in predicting the depth of distant objects. In future work, we will focus on using lightweight network structures to reduce the number of parameters while ensuring the predictive performance of the network and further improving the practicality and efficiency of the network.

REFERENCES

[1] A. Saxena, S. Chung, and A. Ng, "Learning depth from single monocular images," in *Proc. NIPS*, 2005, pp. 1161–1168.
[2] L. Ladicky, J. Shi, and M. Pollefeys, "Pulling things out of perspective," in *Proc. IEEE CVPR*, Columbus, OH, USA, Jun. 2014, pp. 89–96, doi: 10.1109/CVPR.2014.19.
[3] F. Liu, C. Shen, G. Lin, and I. Reid, "Learning depth from single monocular images using deep convolutional neural fields," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 38, no. 10, pp. 2024–2039, Oct. 2016, doi: 10.1109/TPAMI.2015.2505283.
[4] P. Wang, X. Shen, Z. Lin, S. Cohen, B. Price, and A. Yuille, "Towards unified depth and semantic prediction from a single image," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Boston, MA, USA, Jun. 2015, pp. 2800–2809, doi: 10.1109/CVPR.2015.7298897.
[5] D. Eigen and R. Fergus, "Predicting depth, surface normals and semantic labels with a common multi-scale convolutional architecture," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Santiago, Chile, Dec. 2015, pp. 2650–2658, doi: 10.1109/ICCV.2015.304.
[6] R. Garg, V. K. Bg, G. Carneiro, and I. Reid, "Unsupervised CNN for single view depth estimation: Geometry to the rescue," in *Proc. IEEE ICCV*, Amsterdam, The Netherlands, Oct. 2016, pp. 740–756, doi: 10.1007/978-3-319-46484-8_45.
[7] I. Laina, C. Rupprecht, V. Belagiannis, F. Tombari, and N. Navab, "Deeper depth prediction with fully convolutional residual networks," in *Proc. 4th Int. Conf. 3D Vis.*, Stanford, CA, USA, 2016, pp. 239–248, doi: 10.1109/3DV.2016.32.
[8] J. Xie, R. Girshick, and A. Farhadi, "Deep3D: Fully automatic 2D-to-3D video conversion with deep convolutional neural networks," in *Proc. ECCV*, Amsterdam, The Netherlands, 2016, pp. 842–857, doi: 10.1007/978-3-319-46493-0_51.
[9] K. M. He, X. Y. Zhang, S. Q. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. CVPR*, Las Vegas, NV, USA, 2016, pp. 770–778, doi: 10.1109/CVPR.2016.90.
[10] G. Huang, Z. Liu, L. Van Der Maaten, and K. Q. Weinberger, "Densely connected convolutional networks," in *Proc. IEEE CVPR*, Honolulu, HI, USA, Jul. 2017, pp. 2261–2269, doi: 10.1109/CVPR.2017.243.
[11] S. Xie, R. Girshick, P. Dollár, Z. Tu, and K. He, "Aggregated residual transformations for deep neural networks," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Honolulu, HI, USA, Jul. 2017, pp. 5987–5995, doi: 10.1109/CVPR.2017.634.
[12] D. Hoiem, A. A. Efros, and M. Hebert, "Recovering surface layout from an image," *Int. J. Comput. Vis.*, vol. 75, no. 1, pp. 151–172, Jul. 2007, doi: 10.1007/s11263-006-0031-y.
[13] J. Flynn, I. Neulander, J. Philbin, and N. Snavely, "DeepStereo: Learning to predict new views from the world's imagery," 2015, *arXiv:1506.06825*.
[14] K. Karsch, C. Liu, and S. B. Kang, "Depth extraction from video using non-parametric sampling," 2020, *arXiv:2002.04479*.
[15] J. L. Herrera, C. R. Del-Blanco, and N. García, "Automatic depth extraction from 2D images using a cluster-based learning framework," *IEEE Trans. Image Process.*, vol. 27, no. 7, pp. 3288–3299, Jul. 2018, doi: 10.1109/TIP.2018.2813093.
[16] R. Alp Guler, Y. Zhou, G. Trigeorgis, E. Antonakos, P. Snape, S. Zafeiriou, and I. Kokkinos, "DenseReg: Fully convolutional dense shape regression in-the-wild," 2018, *arXiv:1803.02188*.
[17] Y. K. Gan, X. Y. Xu, and W. X. Sun, "Monocular depth estimation with affinity, vertical pooling, and label enhancement," in *Proc. ECCV*, Munich, Germany, 2018, pp. 232–247, doi: 10.1007/978-3-030-01219-9.
[18] D. Eigen, C. Puhrsch, and R. Fergus, "Depth map prediction from a single image using a multi-scale deep network," in *Proc. NIPS*, 2014, pp. 2366–2374.
[19] D. Xu, E. Ricci, W. Ouyang, X. Wang, and N. Sebe, "Multi-scale continuous CRFs as sequential deep networks for monocular depth estimation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Honolulu, HI, USA, Jul. 2017, pp. 161–169, doi: 10.1109/CVPR.2017.25.

[20] W. Yuan, X. Gu, Z. Dai, S. Zhu, and P. Tan, "Neural window fully-connected CRFs for monocular depth estimation," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2022, pp. 3906–3915, doi: 10.1109/CVPR52688.2022.00389.

[21] J. Han Lee, M.-K. Han, D. Wook Ko, and I. Hong Suh, "From big to small: Multi-scale local planar guidance for monocular depth estimation," 2019, *arXiv:1907.10326*.

[22] M. Song, S. Lim, and W. Kim, "Monocular depth estimation using Laplacian pyramid-based depth residuals," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 31, no. 11, pp. 4381–4393, Nov. 2021, doi: 10.1109/TCSVT.2021.3049869.

[23] H. Fu, M. Gong, C. Wang, K. Batmanghelich, and D. Tao, "Deep ordinal regression network for monocular depth estimation," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Salt Lake City, UT, USA, Jun. 2018, pp. 2002–2011, doi: 10.1109/CVPR.2018.00214.

[24] L.-C. Chen, G. Papandreou, I. Kokkinos, K. Murphy, and A. L. Yuille, "DeepLab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected CRFs," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 40, no. 4, pp. 834–848, Apr. 2018, doi: 10.1109/TPAMI.2017.2699184.

[25] Y. Duan, X. Guo, and Z. Zhu, "DiffusionDepth: Diffusion denoising approach for monocular depth estimation," 2023, *arXiv:2303.05021*.

[26] S. Lee, J. Lee, B. Kim, E. Yi, and J. Kim, "Patch-wise attention network for monocular depth estimation," in *Proc. AAAI Conf. Artif. Intell.*, May 2021, vol. 35, no. 3, pp. 1873–1881, doi: 10.1609/aaai.v35i3.16282.

[27] R. Ranftl, A. Bochkovskiy, and V. Koltun, "Vision transformers for dense prediction," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Montreal, QC, Canada, Oct. 2021, pp. 12159–12168, doi: 10.1109/ICCV48922.2021.01196.

[28] A. Agarwal and C. Arora, "Attention attention everywhere: Monocular depth prediction with skip attention," in *Proc. IEEE/CVF Winter Conf. Appl. Comput. Vis. (WACV)*, Waikoloa, HI, USA, Jan. 2023, pp. 5850–5859, doi: 10.1109/WACV56688.2023.00581.

[29] G. Yang, H. Tang, M. Ding, N. Sebe, and E. Ricci, "Transformer-based attention networks for continuous pixel-wise prediction," in *Proc. IEEE ICCV*, Montreal, QC, Canada, Oct. 2021, pp. 16249–16259, doi: 10.1109/ICCV48922.2021.01596.

[30] V. Patil, C. Sakaridis, A. Liniger, and L. Van Gool, "P3Depth: Monocular depth estimation with a piecewise planarity prior," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, New Orleans, LA, USA, Jun. 2022, pp. 1600–1611, doi: 10.1109/CVPR52688.2022.00166.

[31] C. Liu, S. Kumar, S. Gu, R. Timofte, and L. Van Gool, "VA-DepthNet: A variational approach to single image depth prediction," 2023, *arXiv:2302.06556*.

[32] S. Shao, Z. Pei, W. Chen, R. Li, Z. Liu, and Z. Li, "URCDC-depth: Uncertainty rectified cross-distillation with CutFlip for monocular depth estimation," 2023, *arXiv:2302.08149*.

[33] S. Farooq Bhat, I. Alhashim, and P. Wonka, "AdaBins: Depth estimation using adaptive bins," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2021, pp. 4008–4017, doi: 10.1109/CVPR46437.2021.00400.

[34] Z. Li, X. Wang, X. Liu, and J. Jiang, "BinsFormer: Revisiting adaptive bins for monocular depth estimation," 2022, *arXiv:2204.00987*.

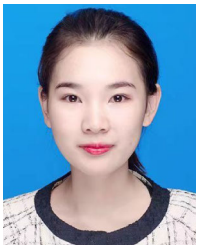
[35] Z. Li, Z. Chen, X. Liu, and J. Jiang, "DepthFormer: Exploiting long-range correlation and local information for accurate monocular depth estimation," *Mach. Intell. Res.*, vol. 20, no. 6, pp. 837–854, Dec. 2023, doi: 10.1007/s11633-023-1458-0.

[36] A. Geiger, P. Lenz, C. Stiller, and R. Urtasun, "Vision meets robotics: The KITTI dataset," *Int. J. Robot. Res.*, vol. 32, no. 11, pp. 1231–1237, Sep. 2013, doi: 10.1177/0278364913491297.

[37] N. Silberman, D. Hoiem, and P. Kohli, "Indoor segmentation and support inference from RGBD images," in *Proc. ECCV*, Florence, Italy, 2012, pp. 746–760, doi: 10.1007/978-3-642-33715-4.

[38] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," 2014, *arXiv:1412.6980*.

[39] M. Sandler, A. Howard, M. Zhu, A. Zhmoginov, and L.-C. Chen, "MobileNetV2: Inverted residuals and linear bottlenecks," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Salt Lake City, UT, USA, Jun. 2018, pp. 4510–4520, doi: 10.1109/CVPR.2018.00474.



JIEJIE XIAO received the degree from Hebei Normal University of Science and Technology, Qinhuangdao, China, in 2020. She is currently pursuing the master's degree with Hebei University of Engineering, Handan, China. Her research interests include image processing based on deep learning and depth estimation in computer vision.

LIHONG LI received the bachelor's degree in mechanical and electronic engineering and the master's and Ph.D. degrees in mechanical manufacturing and automation from Hebei University of Technology, in 1997, 2003, and 2013, respectively. She is currently a Professor with Hebei University of Engineering. She has participated in more than ten national and provincial projects and ten municipal and department-level projects. She has published more than 20 retrieval and core journal articles. Her main research interests include image processing and computer vision.

XU SU received the degree from Hebei University of Engineering, Handan, China, in 2021, where she is currently pursuing the master's degree. Her research interests include image processing based on deep learning and semantic segmentation in computer vision.

GUOPENG TAN received the degree from Yangtze Normal University, Chongqing, China, in 2020. He is currently pursuing the master's degree with Hebei University of Engineering, Handan, China. His research interests include object detection based on deep learning and semantic segmentation in computer vision.