A Low Complexity Visual Depth Estimation Model for Edge AI Devices

Yu-Hsuan Lee
Department of Engineering
Yuan-Ze University
No.135, Yuandong Rd., Zhongli Dist., Taoyuan City 320, Taiwan (R.O.C.)
E-mail: yhlee@saturn.yzu.edu.tw

Chen-Hsuan Wen
Department of Engineering
Yuan-Ze University
No.135, Yuandong Rd., Zhongli Dist., Taoyuan City 320, Taiwan (R.O.C.)
E-mail: s1124621@mail.yzu.edu.tw

***Abstract*—This paper proposes a monocular depth estimation algorithm dedicated for intelligent unmanned vehicles, aiming to achieve both high accuracy and low computational cost. The proposed model delicately integrates Convolutional Neural Network (CNN) and Transformer for a low-complexity depth estimation. Experiment results show that this work outperforms existing compact models in terms of parameter count and computational complexity. It requires only 1.5 million parameters and 1.9 billion floating point operations (FLOPs), while delivering superior performance on multiple standard evaluation metrics. Consequently, this work enables real-time depth estimation to be more feasible in edge AI devices, such as intelligent unmanned vehicles.**

Keywords—Monocular Depth Estimation, Self-Supervised Learning, Convolutional Neural Network, Transformer.

I. INTRODUCTION

In the fields of unmanned aerial vehicles (UAVs) and autonomous driving, depth maps are widely used to represent the three-dimensional structure of scenes. Therefore, to obtain depth information in a low-cost manner is a critical issue. Common depth acquisition methods include stereo vision, LiDAR, and monocular depth estimation. Although stereo vision can directly estimate depth from disparity between two cameras, it also suffers from unreliable accuracy at long distances and constraints. In addition, the baseline distance between the two cameras especially on smaller devices, seriously limiting depth resolution. LiDAR systems offer long-range measurement capabilities [1], typically up to around 100 meters, and can extend this range by increasing the emission power. However, this technology requires active laser emission, which leads to considerable power consumption in its functional modules. In contrast, monocular depth estimation predicts depth from a single image, resulting in lower hardware costs and greater applicability. As a result, it attracts considerable research effort in recent years.

Fig. 1 illustrates a monocular depth estimation application scenario. First, a single RGB image is captured using a monocular camera, and an AI model is then employed to estimate the depth map for each pixel in the image. With the depth information obtained, the system can estimate the relative distance between objects in the scene and the camera, thereby enabling a wide range of perception and control tasks. Building upon this capability, researchers have explored diverse applications. For instance, a lightweight drone was designed to achieve obstacle avoidance using only a monocular camera, eliminating the need for additional depth sensors. That makes it suitable for resource-constrained environments [2]. In the context of

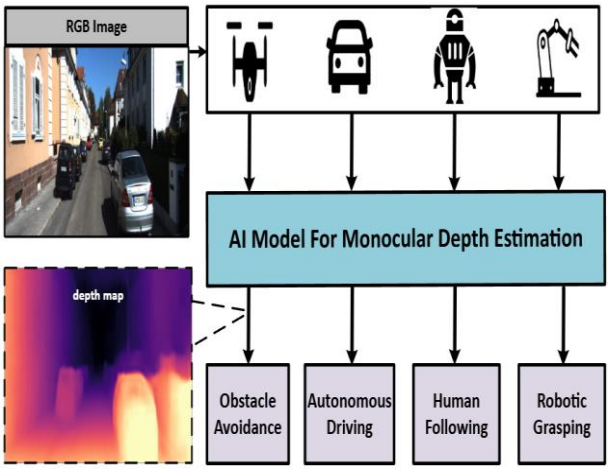

Fig. 1. Monocular Depth Estimation Scenario Diagram

autonomous driving, Instance Clustering Guidance has been introduced to enhance monocular depth estimation accuracy, improving the vehicle's ability to avoid dynamic obstacles such as pedestrians and other cars [3]. Similarly, in mobile robotics, depth estimation has been integrated with object tracking to enable robots to follow humans and navigate safely in complex environments, relying solely on RGB images [4]. Moreover, monocular vision has also been applied to robotic manipulation, where a four degree of freedom (4 DoF) robotic arm leverages depth estimation to measure distances between itself and surrounding objects [5]. These examples collectively highlight the versatility of monocular depth estimation and its potential to support intelligent perception in a variety of real-world scenarios.

However, for all of these applications, endurance is a key consideration. For example, small drones or mobile robots require energy-relay and energy-saving strategies to extend operation time during long-term or continuous patrol. Therefore, when designing such systems, it is generally necessary to develop lightweight models with low parameter counts and FLOPs to reduce computational load and power consumption. Meanwhile, the accuracy of depth estimation should be also acceptable.

Recently, an increasing number of computer vision tasks have leveraged hybrid architectures that combine Convolutional Neural Networks (CNNs) with Transformers [6], aiming to simultaneously capture local details and long-range global dependencies. For instance, the Convolution-Enhanced Image Transformer (CeiT) integrates convolutional operations into the Transformer structure and achieves strong performance on various ImageNet benchmarks [7][8]. Similarly, CoAtNet unifies depthwise

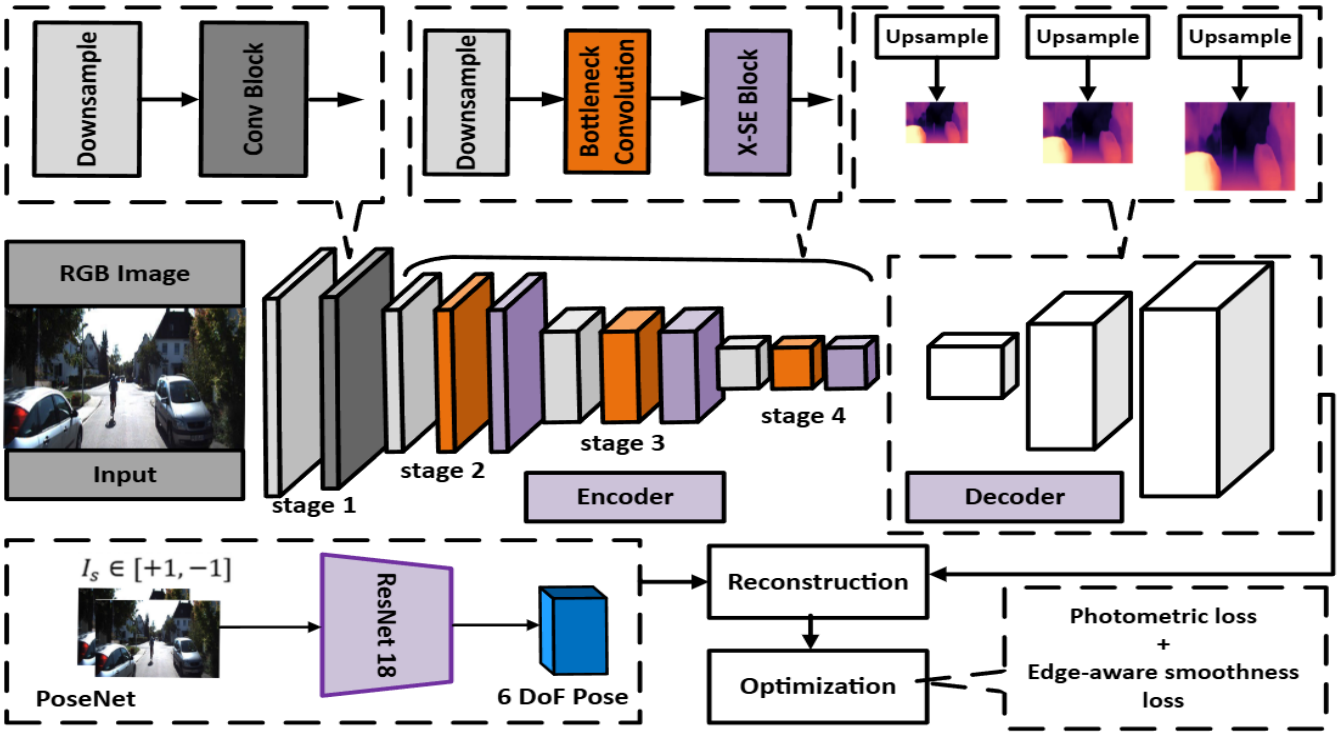

Fig. 2. Overall Architecture

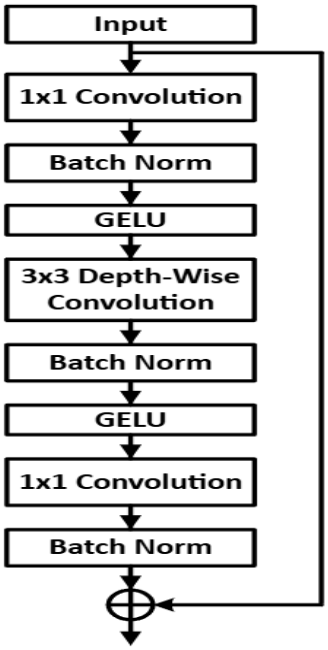

Fig. 3. Improved Bottleneck Convolutions

convolution with self-attention, yielding high accuracy in large-scale image classification tasks [9]. In the medical domain, TransUNet incorporates the strengths of both Transformers and U-Net, enabling effective performance across diverse medical image analysis applications [10]. These advances collectively highlight the complementary nature of CNNs and Transformers, which has gradually evolved into a significant research direction in computer vision.

Inspired by these approaches, this work designed a monocular depth estimation model based on CNN and Transformer. It is trained with a self-supervised learning method using image reconstruction error as the learning signal. Hence, it is unnecessary to pay more expensive and difficult burdens to obtain ground truth depth labels. This work can reduce the number of parameters while maintaining model accuracy. With this work, an efficient depth estimation can be feasible especially for resource-constrained edge devices.

The remainder of this paper is organized as follows. The proposed methodology is reported in Section II. Experiment results and discussion are presented in Section III. Finally, conclusions are given in Section IV.

II. Proposed Methodology

Generally, edge devices are often deployed under resource-limited conditions, and thereby a lightweight backbone network is required. Balancing performance and model size becomes a critical issue. Fig. 2 shows the overall architecture of proposed model, which is a hybrid of CNN and Transformer. An encoder is used to learn and extract multiscale features. Then, a decoder generates inverse depth maps at different resolutions, accompanied by a PoseNet module for pose estimation. Subsequently, a reconstructed target image is produced, and self-supervised learning is applied to compute the loss and optimize the model. Each module is sequentially reported in the following subsections.

A. Encoder

To effectively capture features at different levels, this work adopts the same four-stage multi-scale feature aggregation strategy as Lite-Mono [11], enhancing feature richness and decoding performance. The encoder first takes a single image of resolution $H \times W$ and puts it into stage 1. Through downsampling and a 3×3 convolution, it extracts features resulting in a feature map of size $H/2 \times W/2 \times C_1$. These features are then concatenated with a pooled version of the input image and fed into stage 2. In stage 2, downsampling is applied again, producing a feature map of size$H/4 \times W/4 \times C_2$. Several improved Bottleneck

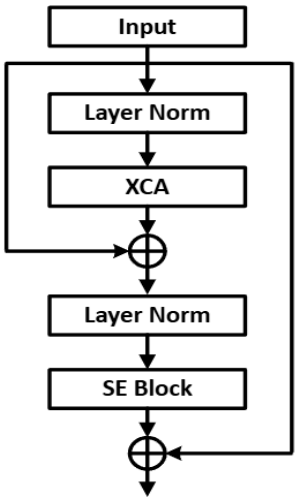

Fig. 4. Proposed X-SE Block

Convolutions [12] are then used to learn features, followed by the application of an X-SE Block module which combines Cross-Covariance Attention (XCA) [13] and Squeeze-and-Excitation (SE) [14], strengthening inter-channel interactions and selective emphasis on important features. Stage 3 and stage 4 employ the same approach, generating feature maps sized $H/8 \times W/8 \times C_3$ and $H/16 \times W/16 \times C_4$, respectively.

Depthwise separable convolution is a lightweight strategy that decomposes a standard convolution into depthwise convolution and pointwise convolution. Depthwise convolution performs spatial convolution independently on each input channel, resulting in the same number of output channels as input channels without mixing information across channels during computation.

Although this design greatly reduces computational cost, it also has a limitation as follows. It restricts the free expansion of channel numbers during feature extraction, thereby limiting the model's representational capacity. Performing convolution only in a low-dimensional space can reduce computational cost but insufficient channels may degrade semantic feature representation.

To balance computational efficiency and feature expressiveness, this paper employs the Bottleneck Convolutions proposed in MobileNetV2 for local feature extraction. Specifically, a 1×1 convolution first expands the channel dimension to extract richer semantic features. Then, Depthwise Convolution performs feature extraction in the high-dimensional space. Finally, another 1×1 convolution maps the features back to the original dimension, combining the feature maps produced across channels. The module's output is added to its input via a residual connection and passed to subsequent network layers. Additionally, we replace the activation function from ReLU to GELU[15]. Since ReLU assigns zero for all inputs less than zero, it may discard certain useful negative-valued features. In contrast, GELU preserves part of the negative information in a smooth manner, enabling the model to better exploit the input features. More detail of this architecture is shown in Fig. 3.

Fig. 4 shows the proposed X-SE Block. To overcome the limitation of convolutional operations that primarily capture local information, it incorporates Cross-Covariance Attention (XCA), which enhances feature representation by capturing global dependencies across channels. Given an input feature map X of size $H \times W$ with C channels, it is linearly projected into three distinct components: queries($Q = XW_Q$), keys ($K = XW_K$), and values ($V = XW_V$), where W_Q, W_K, and W_V are weight matrices. The Cross-Covariance Attention is then applied to enhance the input features X, as formulated in (1):

$$X' = XCA(Q,K,V) + X \tag{1}$$

where, $XCA(Q,K,V)$ is derived by multiplying the transposed key matrix K^T with the query matrix Q. It adopts Softmax function to generate attention weights, and then multiplies these weights with the values matrix V, as expressed in (2):

$$XCA(Q,K,V) = V \cdot Softmax(K^T \cdot Q) \tag{2}$$

Furthermore, this work integrates the Squeeze-and-Excitation (SE) mechanism to enhance the model's responsiveness to critical channels. In the Squeeze stage, global average pooling is applied to the feature maps to generate a descriptor vector that represents the overall information of each channel. In the Excitation stage, a fully connected layer first reduces the dimension from C to C/r, where r denotes the reduction ratio. This is followed by a ReLU activation to enhance non-linear representation. Another fully connected layer then expands the dimension back from C/r to C, and then, a sigmoid function is applied to obtain the weights of each channel. Finally, each channel feature is multiplied by its corresponding weight, thereby strengthening important channels while suppressing redundant ones.

B. Decoder

This work adopts a decoder architecture as Lite-Mono, using bilinear upsampling to progressively enlarge feature maps, and fuse features from the corresponding encoder layers at each stage. Subsequently, inverse depth maps are output at 1/4, 1/2, and full resolutions.

C. PoseNet

This work chooses ResNet18 as the PoseNet, similar to [11][16][17], taking three consecutive frames as input to predict the six degrees of freedom (6-DoF) relative pose among the three frames.

D. Self-Supervised Learning

Self-supervised learning enables training in the absence of ground truth depth annotations. Instead, we use image reconstruction as the supervisory signal, following [11] and [16]. Training is performed by projecting images into adjacent views and minimizing their photometric reprojection loss. Similar to [11] and [16], this work uses an edge-aware smoothness loss to encourage smoother inverse depth predictions.

First, the reconstructed target image $I_t{}'$ from the target image I_t is obtained as:

$$I_t{}' = \theta(I_s, K, T_{t\to s}, D_t) \tag{3}$$

where θ denotes the reconstruction function. I_s is the source image, which can be the previous or next frame to I_t. K is the camera intrinsic matrix. $T_{t\to s}$ is the relative pose estimated by the PoseNet. D_t is the depth predicted by the network. Following [11] and [16], this work introduces the photometric reprojection loss L_{re} composed of an L_1 loss

TABLE I. Results after evaluation on the KITTI dataset (resolution: 640×192)

Method	Depth Error(↓)				Depth Accuracy(↑)		
	Abs Rel	Sq Rel	RMSE	RMSE log	δ_1 <1.25	δ_2 <1.25^2	δ_3 <1.25^3
R-MSFM3[22]	0.114	0.815	4.712	0.193	0.876	0.959	0.981
R-MSFM6[22]	0.112	0.806	4.704	0.191	0.878	0.960	0.981
Lite-mono-small[11]	0.110	0.802	4.671	0.186	0.879	0.961	0.982
Lite-mono-tiny[11]	0.110	0.837	4.710	0.187	0.880	0.960	0.982
Ours	**0.109**	**0.792**	**4.648**	**0.184**	**0.882**	**0.961**	**0.983**

TABLE II. Results after evaluation on the KITTI dataset (resolution: 1024×320)

Method	Depth Error(↓)				Depth Accuracy(↑)		
	Abs Rel	Sq Rel	RMSE	RMSE log	δ_1 <1.25	δ_2 <1.25^2	δ_3 <1.25^3
R-MSFM3[22]	0.112	0.773	4.581	0.189	0.879	0.960	0.982
R-MSFM6[22]	0.108	0.748	4.470	0.185	0.889	0.963	0.982
Lite-mono-small[11]	0.103	0.757	4.449	0.180	0.894	0.964	0.983
Lite-mono-tiny[11]	0.104	0.764	4.487	0.180	0.892	0.964	0.983
Ours	**0.107**	**0.764**	**4.524**	**0.183**	**0.887**	**0.963**	**0.983**

and Structural Similarity Index Measure (SSIM):

$$L_{re}(I_t, I_t') = \frac{\alpha}{2} \cdot 1 - SSIM(I_t, I_t') + (1-\alpha) \cdot \|I_t - I_t'\|_1 \quad (4)$$

where α=0.85 [16]. We also use the per-pixel minimum reprojection loss L_P [16] to handle pixels outside the field of view and occluded regions:

$$L_P(I_s, I_t) = \min_{I_s \in [-1,1]} L_{re}(I_t, I_t') \quad (5)$$

Additionally, a binary mask μ [16] is applied to filter out pixels in the frame sequence whose appearance remains unchanged:

$$\mu = \min_{I_s \in [-1,1]} L_{re}(I_s, I_t) > \min_{I_s \in [-1,1]} L_{re}\left(I_t, I_t'\right) \quad (6)$$

As in [11] and [16], an edge-aware smoothness loss L_S is used to encourage smoother inverse depth predictions:

$$L_S = |\partial_x d_t^*| e^{-|\partial_x I_t|} + |\partial_y d_t^*| e^{-|\partial_y I_t|} \quad (7)$$

where $d_t^* = d_t / \overline{d_t}$ is the mean-normalized inverse depth. The final loss L is computed by combining the losses across multiple scales:

$$L = \frac{1}{3} \sum_{s \in \left\{1, \frac{1}{2}, \frac{1}{4}\right\}} (\mu L_P + \lambda L_S) \quad (8)$$

where s denotes the different output scales of the decoder, and λ is the weight for the smoothness loss, set to $1e^{-3}$ as indicated in [11].

III. Experiment Results and Discussion

A. Dataset

This experiment uses the KITTI [18] dataset and adopt the Eigen splits [19] for training and evaluation. The dataset consists of a total of 39,180 monocular triplets for training, 4,424 for validation, and 697 for testing.

B. Implementation Details

This work is implemented using PyTorch and trained on a single NVIDIA 3090 GPU for 30 epochs with a batch size of 12. The optimizer is AdamW [20], with an initial learning rate of $1e^{-4}$. To accelerate convergence, we first pre-train the backbone on the ImageNet1K dataset [8] for 100 epochs. The pre-training is conducted using Distributed Data Parallel (DDP) with 2 GPUs, where each GPU uses a batch size of 256.

To ensure good generalization performance, we adopt the same data augmentation strategies as [11][16]. Specifically, each augmentation is applied with a 50% chance: including horizontal flipping, random saturation adjustment (±0.2), random brightness adjustment (±0.2), random contrast adjustment (±0.2), and hue jitter (±0.1).

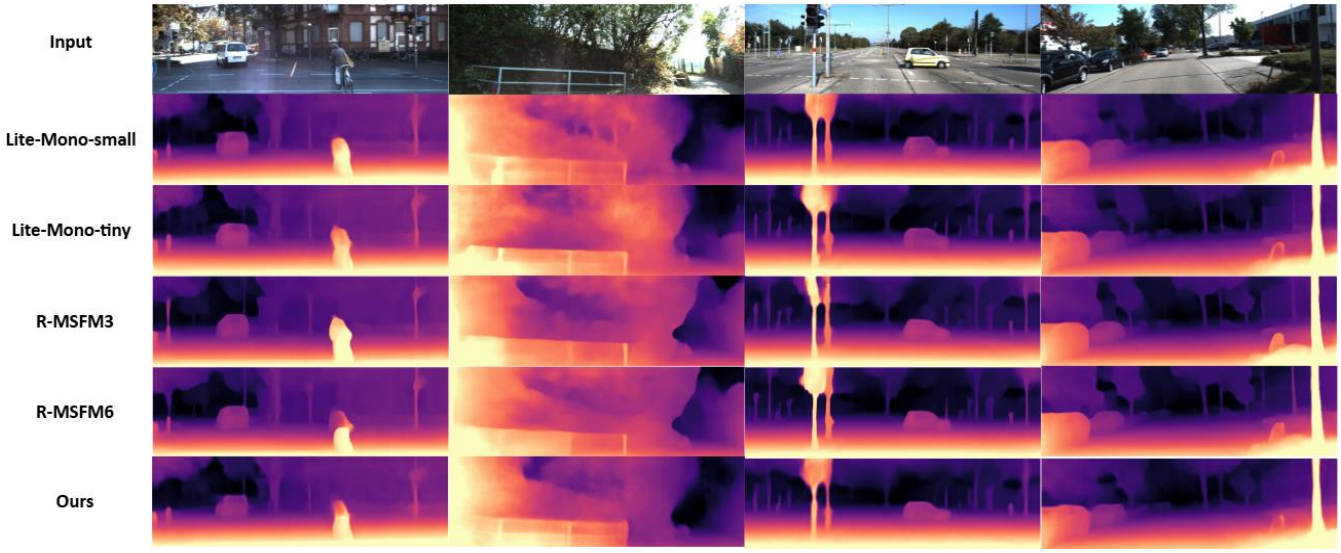

Fig. 5. Visualization results.

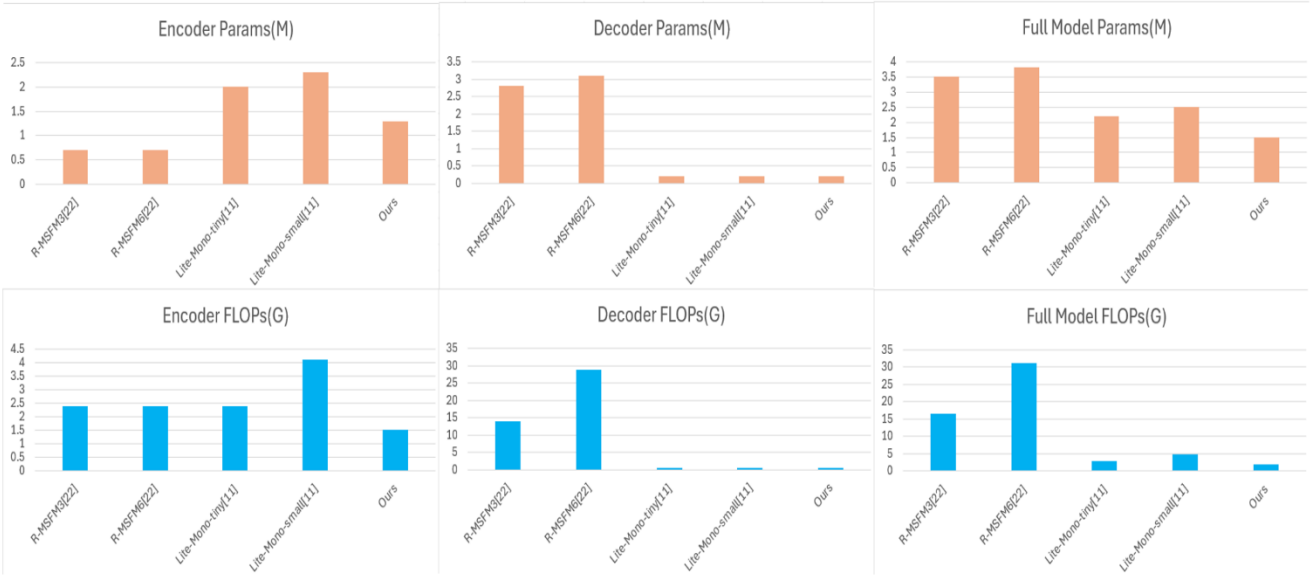

Fig. 6. Model Complexity Comparison

C. *Evaluation Results on Dataset*

The proposed model is evaluated on the KITTI dataset using images with resolutions of 640×192 and 1024×320. The evaluation follows the metrics proposed in [21]. The metrics are divided into two categories: Depth Error (the lower, the better) and Depth Accuracy (the higher, the better). Depth Error includes four indicators: Absolute Relative Error (Abs Rel), Squared Relative Error (Sq Rel), Root Mean Squared Error (RMSE), and RMSE log. Depth Accuracy is measured by the thresholds δ_1 <1.25, δ_2 <1.25^2, δ_3 <1.25^3.

Experiment results are shown in TABLE I and TABLE II. At a resolution of 640×192, we compared our model with other sophisticated models. The results show that this work can achieve better performance across all metrics. Compared with R-MSFM3 [22], our model performed better in every metric, and even when compared to the larger R-MSFM6 [22], this work still reaches better performance. Furthermore, compared to the more advanced small models in recent years—Lite-Mono-small [11] and Lite-Mono-tiny [11], this work achieved lower depth error and higher depth accuracy, while also keeping smaller model size than all of the above models.

At a resolution of 1024×320, this model is also superior to those of R-MSFM3. Although slightly inferior to R-MSFM6, Lite-Mono-small and Lite-Mono-tiny, but our approach just requires lower parameter counts and less FLOPs, exhibiting higher computation-efficiency. Fig. 5 shows the visualization results. Although this work demands fewer parameters, the generated depth maps can still clearly depict object contours.

D. *Model Complexity Analysis*

Fig. 6 compares the model complexity of our method with other approaches, focusing on the number of parameters and FLOPs. The top three charts individually show the parameter counts of the encoder, decoder, and the entire model. The bottom three charts are for the FLOPs of the encoder, decoder, and the entire model, respectively.

From the bar charts, it can be seen that our encoder has about 1.3M parameters, which is less than both Lite-Mono-small and Lite-Mono-tiny. Although it is slightly larger

than R-MSFM3 and R-MSFM6, our decoder adopts a Lite-Mono-like architecture, which contains significantly fewer parameters compared to R-MSFM3 and R-MSFM6.

Overall, the proposed model achieves the least parameters of about 1.5M, approximately 40% less than Lite-Mono-small and about 60% fewer than R-MSFM6. In terms of FLOPs, this model reaches about 1.9G, around 60% lower than Lite-Mono-small and up to 94% lower than R-MSFM6.

IV. Conclusion

In this paper, a lightweight model architecture is proposed. Experimental results show that this model requires only 1.5M parameters and 1.9G FLOPs. Both are significantly lower than those of other competitive small-scale models. Moreover, this model demonstrates superior performance on various metrics. Due to its lightweight design and low computational burdens, our approach has strong deployment potential on resource-constrained edge devices. Compared with large-scale models, this work is more efficient to be implemented on hardware especially for real-time depth estimation.

References

[1] S. Zhuo *et al*., "Solid-State dToF LiDAR System Using an Eight-Channel Addressable, 20-W/Ch Transmitter, and a 128 × 128 SPAD Receiver With SNR-Based Pixel Binning and Resolution Upscaling," in *IEEE Journal of Solid-State Circuits*, vol. 58, no. 3, pp. 757-770, Mar. 2023.

[2] G. Higashiuchi, H. Nishikawa, X. Kong and H. Tomiyama, "Obstacle Avoidance using Monocular Depth Estimation for Small Drone Tello," *2024 International Technical Conference on Circuits/Systems, Computers, and Communications (ITC-CSCC)*, Okinawa, Japan, 2024, pp. 1-6.

[3] D. Kim, D. Jin and C. -S. Kim, "Monocular Depth Estimation for Autonomous Driving Based on Instance Clustering Guidance," *2024 Asia Pacific Signal and Information Processing Association Annual Summit and Conference (APSIPA ASC)*, Macau, Macao, 2024, pp. 1-6.

[4] T. -H. Tsai and C. -L. Lee, "Equipped with Monocular Depth Estimation and Intelligent Wake-Up Vision Based Tracking System for a Human-Following Mobile Robot," *2024 IEEE International Conference on Multimedia and Expo Workshops (ICMEW)*, Niagara Falls, ON, Canada, 2024, pp. 1-2.

[5] E. Chebotareva, A. Mukhamedshin, N. Imamov and E. Magid, "Object Localization Based on a Single RGB Camera for a 4-DOF Robotic Arm," *2025 11th International Conference on Automation, Robotics, and Applications (ICARA)*, Zagreb, Croatia, 2025, pp. 252-256.

[6] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin, "Attention Is All You Need," 2017, *arXiv:1706.03762.*

[7] K. Yuan, S. Guo, Z. Liu, A. Zhou, F. Yu and W. Wu, "Incorporating Convolution Designs into Visual Transformers," *2021 IEEE/CVF International Conference on Computer Vision (ICCV)*, Montreal, QC, Canada, 2021, pp. 559-568.

[8] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *CVPR*, 2009.

[9] Zihang Dai, Hanxiao Liu, Quoc V. Le, and Mingxing Tan, "CoAtNet: Marrying Convolution and Attention for All Data Sizes," 2021, *arXiv:2106.04803*.

[10] Jieneng Chen, Yongyi Lu, Qihang Yu, Xiangde Luo, Ehsan Adeli, Yan Wang, Le Lu, Alan L. Yuille, and Yuyin Zhou, "TransUNet: Transformers Make Strong Encoders for Medical Image Segmentation," 2021, *arXiv:2102.04306.*

[11] N. Zhang, F. Nex, G. Vosselman and N. Kerle, "Lite-Mono: A Lightweight CNN and Transformer Architecture for Self-Supervised Monocular Depth Estimation," *2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, Vancouver, BC, Canada, 2023, pp. 18537-18546.

[12] M. Sandler, A. Howard, M. Zhu, A. Zhmoginov and L. -C. Chen, "MobileNetV2: Inverted Residuals and Linear Bottlenecks," *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, Salt Lake City, UT, USA, 2018, pp. 4510-4520.

[13] Alaaeldin Ali, Hugo Touvron, Mathilde Caron, Piotr Bojanowski, Matthijs Douze, Armand Joulin, Ivan Laptev, Natalia Neverova, Gabriel Synnaeve, Jakob Verbeek, et al. Xcit: Cross-covariance image transformers. *NeurIPS*, 2021.

[14] J. Hu, L. Shen and G. Sun, "Squeeze-and-Excitation Networks," *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, Salt Lake City, UT, USA, 2018, pp. 7132-7141.

[15] Dan Hendrycks and Kevin Gimpel. Bridging nonlinearities and stochastic regularizers with gaussian error linear units. *CoRR*, 2016

[16] C. Godard, O. M. Aodha, M. Firman and G. Brostow, "Digging Into Self-Supervised Monocular Depth Estimation," *2019 IEEE/CVF International Conference on Computer Vision (ICCV)*, Seoul, Korea (South), 2019, pp. 3827-3837.

[17] Z. Liu, R. Li, S. Shao, X. Wu and W. Chen, "Self-Supervised Monocular Depth Estimation With Self-Reference Distillation and Disparity Offset Refinement," in *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 33, no. 12, pp. 7565-7577, Dec. 2023.

[18] Andreas Geiger, Philip Lenz, Christoph Stiller, and Raquel Urtasun. Vision meets robotics: The kitti dataset. *Int. J. Robot. Res*., 32(11):1231–1237, 2013.

[19] David Eigen and Rob Fergus. Predicting depth, surface normals and semantic labels with a common multi-scale convolutional architecture. In *ICCV*, 2015.

[20] Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. In *ICLR*, 2018

[21] D. Eigen, C. Puhrsch, and R. Fergus, "Depth map prediction from a single image using a multi-scale deep network," in *Proc. Adv. Neural Inf. Process. Syst*., vol. 27, 2014, pp. 1–9.

[22] Z. Zhou, X. Fan, P. Shi and Y. Xin, "R-MSFM: Recurrent Multi-Scale Feature Modulation for Monocular Depth Estimating," *2021 IEEE/CVF International Conference on Computer Vision (ICCV)*, Montreal, QC, Canada, 2021, pp. 12757-12766.