Spatial-Aware Dynamic Lightweight Self-Supervised Monocular Depth Estimation

Linna Song, Dianxi Shi, Jianqiang Xia, Qianying Ouyang, Ziteng Qiao, Songchang Jin, and Shaowu Yang

***Abstract*—Self-supervised monocular depth estimation has attracted extensive attention in recent years. Lightweight depth estimation methods are crucial for resource-constrained edge devices. However, existing lightweight methods often encounter the challenge of limited representation capacity and increased computational resource consumption for image reconstruction. To alleviate these issues, we propose a novel spatial-aware dynamic lightweight monocular depth estimation method (SAD-Depth). Specifically, we propose a spatial-aware dynamic encoder, which can capture spatial information of the input and generate input-adaptive dynamic convolutions, thereby significantly enhancing the model's adaptability to complex scenes. Meanwhile, we propose a multi-scale sub-pixel lightweight decoder that generates high-quality depth maps while maintaining a lightweight design. Experimental results demonstrate that our proposed SAD-Depth exhibits superiority in both model size and inference speed, achieving state-of-the-art performance on the KITTI benchmark.**

***Index Terms*—Deep learning for visual perception, mapping, lightweight methods.**

I. Introduction

DEPTH estimation is a core task in computer vision that leverages data (e.g., images, point clouds, etc.) acquired from sensor devices (e.g., cameras, LiDAR, etc.) to infer scene depth information. For edge devices, obtaining accurate depth information of the scene is of paramount importance in various domains, such as autonomous driving [1], localization and navigation [2], and augmented reality [3]. Compared to costly depth sensors such as lidar, stereo cameras, and depth cameras, monocular cameras have the advantages of low cost, easy deployment, and low power consumption, which make them the first choice for resource-constrained platforms such as micro-robots, micro-drones, and other edge devices. Monocular depth estimation methods based on deep learning have made significant progress in recent years. However, the monocular depth estimation methods based on the supervised paradigm require a large number of depth labels [4], [5], which are expensive and difficult to obtain. To address this issue, Zhou et al. [6] presented a self-supervised learning framework for the monocular depth and camera motion estimation from unlabeled monocular videos. Since then, there has been extensive research on self-supervised monocular depth estimation [7], [8], [9].

With the continuous development of edge devices, researchers have proposed lightweight depth estimation methods [10], [12], [13], [14], characterized by reduced model size, fewer parameters, and lower computational complexity. These methods often employ depthwise separable convolutions [15] to reduce the number of parameters and computational complexity. Compared with standard convolutions, depthwise separable convolutions exhibit a reduction in both parameters and computations, potentially resulting in diminished feature extraction capacity and subsequently impacting the performance of the model. Furthermore, increasing the resolution of image reconstruction raises the computational burden on the decoder due to the need for processing more pixels and involving complex convolution operations.

To address the challenges of limited representation capability and increased computational resource consumption for image reconstruction in lightweight depth estimation models, we propose a spatial-aware dynamic lightweight monocular depth estimation method (SAD-Depth). To tackle the problem of weak representation capability in lightweight depth estimation models, we propose a spatial-aware dynamic encoder (SAD-Encoder) with spatial-aware dynamic depthwise separable convolution (SAD-DS) modules. SAD-DS can enhance the representation capacity of depthwise separable convolutions by generating input-adaptive dynamic depthwise convolution kernels based on the spatial information aggregated from the inputs, resulting in the acquisition of richly informative features. Secondly, to reduce the model's size while maintaining high-resolution depth image reconstruction, we design a multi-scale sub-pixel lightweight decoder (SUB-Decoder). Sub-pixel convolution [16] is an upsampling technique that enables precise restoration of high-resolution images. Simultaneously, sub-pixel upsampling reduces the number of feature channels, effectively alleviating the computational burden of subsequent convolution operations. Our proposed SAD-Depth, with only 2.6 M parameters, demonstrates exceptional performance.

Our main contributions can be summarized as follows:

Manuscript received 23 July 2023; accepted 20 November 2023. Date of publication 30 November 2023; date of current version 12 December 2023. This letter was recommended for publication by Associate Editor U. Frese and Editor S. Behnke upon evaluation of the reviewers' comments. This work was supported by the Integrated Program of National Natural Science Foundation of China under Grant 91948303. *(Corresponding author: Dianxi Shi.)*

Linna Song and Shaowu Yang are with the College of Computer, National University of Defense Technology, Changsha 410073, China (e-mail: songlinna21@nudt.edu.cn; shaowu.yang@nudt.edu.cn).

Dianxi Shi, Qianying Ouyang, Ziteng Qiao, and Songchang Jin are with the Intelligent Game and Decision Lab, Beijing 100091, China (e-mail: dxshi@nudt.edu.cn; oyqy@nudt.edu.cn; ztqiao99@163.com; jsc04@tsinghua.org.cn).

Jianqiang Xia is with the National Innovation Institute of Defense Technology, Beijing 100071, China (e-mail: jianqiang.xia@foxmail.com).

Digital Object Identifier 10.1109/LRA.2023.3337991

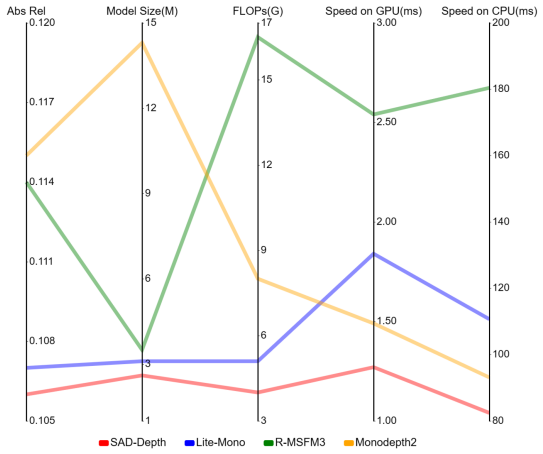


Fig. 1. Performance comparison for various depth estimation methods. Our proposed method SAD-depth outperforms lite-Mono [10], R-MSFM3 [11], and Monodepth2 [7] on metrics Abs Rel, model size, FLOPs, and inference time on GPU and CPU.

- We propose a spatial-aware dynamic encoder in which the convolutional kernels are adaptively generated based on the spatial context of the input, thereby enhancing the feature extraction capability of the encoder and the performance of the model.
- We propose a multi-scale sub-pixel lightweight decoder that is capable of reducing the number of feature channels while maintaining high-resolution image reconstruction, thereby reducing the size of the decoder.
- The experimental results demonstrate that our proposed SAD-Depth outperforms Monodepth2 [7] on the KITTI dataset [17] with 0.106 Abs Rel, while using only 20 % of the trainable parameters. Additionally, SAD-Depth exhibits a significant decrease in inference speed on both GPU and CPU processors, as illustrated in Fig. 1.

II. Related Work

A. Monocular Depth Estimation

Monocular depth estimation is a task to infer depth information from images obtained by monocular cameras. In recent years, deep learning has demonstrated excellent performance in monocular depth estimation tasks. Eigen et al. [4] first proposed a multi-scale monocular depth estimation network to regress depth maps. Considering the inherent ordered relationship among depth values, Cao et al. [5] treated the depth estimation as a pixel-level classification task. These supervised learning methods rely on a large amount of ground truth depth values, which can be costly to obtain. To avoid reliance on ground truth, Zhou et al. [6] proposed a method that utilizes video sequences captured by a monocular camera as input and jointly optimizes the depth estimation network and the pose estimation network. This work demonstrated the feasibility of estimating depth from a single image without ground truth. However, its performance is still influenced by occlusions and moving objects. Godard et al. [7] proposed Monodepth2, which addresses occluded pixels using per-pixel minimum reprojection loss and ignores training pixels that violate the camera motion assumption using an auto-masking loss. To obtain depth information with richer details, Lyu et al. [18] proposed the HR-Depth method. This approach re-designed the skip connections in the depth network and introduced a feature fusion squeeze and excitation module to obtain better high-resolution features.

B. Lightweight Depth Estimation

With the increasing demand for real-time applications on edge devices, researchers have designed a series of lightweight monocular depth estimation networks based on lightweight CNNs such as MobileNets [15], ShuffleNet [19], and ESP-Net [20]. Wofk et al. [12] proposed FastDepth based on depth-wise separable convolutions and applied network pruning to further reduce computational complexity and latency. However, lightweight models may suffer from the loss of fine-grained details and edge blurring in the predicted depth maps. Rudolph et al. [14] proposed guided upsampling blocks for building the decoder to reconstruct high-resolution depth maps. Zhou et al. [11] proposed a recurrent multi-scale feature modulation to gradually refine the depth map. In order to capture global contexts, Zhang et al. [10] proposed a lightweight CNN and transformer hybrid architecture to significantly reduce the model size while maintaining the accuracy of the model.

These lightweight methods face the challenge of limited representation capacity and increased computational resource consumption for image reconstruction. Inspired by CondConv [21], which can dynamically adjust the convolution kernel according to different inputs and sub-pixel convolution [16], which can reconstruct images with rich details, we propose a spatial-aware dynamic lightweight depth estimation network. With this design, we can achieve high accuracy depth estimation at a very low computational cost.

III. Method

In this section, we will present our spatial-aware dynamic lightweight depth estimation method (SAD-Depth). SAD-Depth consists of a spatial-aware dynamic encoder and a multi-scale sub-pixel lightweight decoder, both of which we describe in detail. Subsequently, we introduce our training strategy, which involves self-supervised learning and uncertainty-aware self-teaching methods.

A. Spatial-Aware Dynamic Encoder

1) Structure of Spatial-Aware Dynamic Encoder: The primary task of the encoder is to extract depth information from the input RGB images. However, when dealing with complex and diverse images, the encoder struggles to adequately adapt to various spatial scenarios, which can result in insufficient or inaccurate depth information extraction. To overcome this limitation, we introduce a spatial-aware dynamic encoder that incorporates our designed spatial-aware dynamic depthwise separable convolution modules, as shown in the SAD-Decoder in Fig. 2(a). This design enables the encoder to dynamically

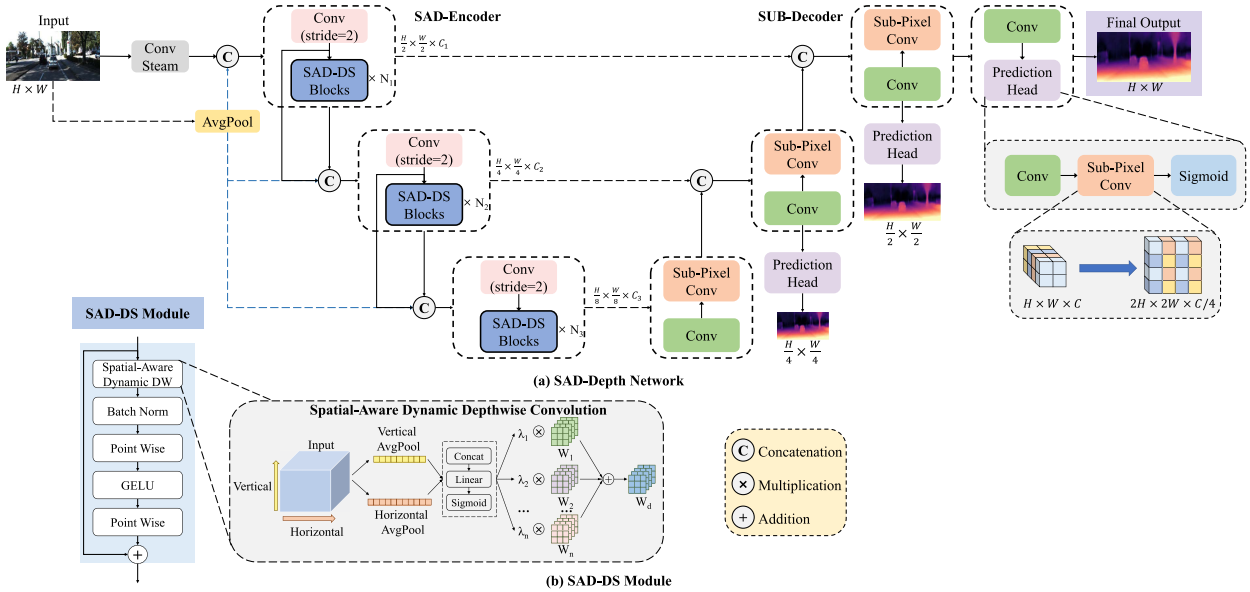


Fig. 2. Architecture of our proposed network. (a) Spatial-aware dynamic lightweight depth estimation (SAD-depth): The encoder composed of SAD-DS extracts multi-scale features, and the decoder composed of sub-pixel outputs multi-scale depth maps. (b) Spatial-aware dynamic depthwise separable convolution (SAD-DS) module: The dynamic convolution kernel is obtained by dynamically aggregating multiple static convolution kernels based on the spatial context of the input.

adapt to images with varying spatial structures, enhancing the feature extraction capability of the encoder and, consequently, significantly boosting the performance of the depth estimation model.

Each stage of the encoder consists of convolution downsampling with stride 2 and our proposed spatial-aware dynamic depthwise separable convolution (SAD-DS) modules. In the first and second stages, there are 3 SAD-DS modules each, while in the third stage, there are 6 SAD-DS modules. To enlarge the receptive field of the encoder, we employ SAD-DS with different dilation rates in each stage to extract multi-scale features. Following Lite-Mono [10], we concatenate the features obtained from the SAD-DS modules with the average-pooled features of the input image and the features obtained from the downsampling layers to reduce spatial information loss due to the reduction in feature scale. In detail, the RGB image with size H × W is initially processed through a convolution stem for preliminary feature extraction, resulting in features with size H/2×W/2. Then, through three stages, features with sizes H/4×W/4, H/8×W/8, and H/16×W/16 are obtained, respectively.

2) Spatial-Aware Dynamic Depthwise Separable Convolution Module: The depthwise separable convolution is a lightweight convolution consisting of a depthwise convolution and a pointwise convolution. Among them, the depthwise convolution reduces the computational cost by sharing the depthwise convolution kernel at the same spatial position in different channels. However, when dealing with inputs that have different spatial information, the spatial weight sharing of depthwise convolution limits its feature extraction capability, resulting in pool feature learning. To enhance the feature learning capability while maintaining the lightweight nature of the depthwise separable convolution, we propose a plug-and-play spatial-aware dynamic depthwise separable convolution (SAD-DS) module, as shown in Fig. 2(b). This module consists of pointwise convolution, GELU activation, batch normalization, and our designed dynamic depthwise convolution.

The dynamic depthwise convolution, as shown in Fig. 2(b), is composed dynamically by multiple convolutional kernels based on the spatial context of the input. Specifically, for the input, we aggregate its spatial information along both the vertical and horizontal directions using global average pooling. By utilizing the spatial information of the input, we obtain dynamic weights λ_i corresponding to different static convolution kernels W_i. We then multiply the dynamic weights λ_i with the static convolution kernels W_i to adaptively generate the dynamic convolution kernels W_d for different inputs. The above process of obtaining the dynamic depthwise convolution kernel can be defined by (1), (2). Given the input x, the weights λ_i corresponding to different static convolution kernels W_i can be obtained by (1), and the dynamic convolution kernel is defined by (2) :

$$\lambda(x) = Sigmoid(Linear(Concat(\mathrm{GAP}_V(x), \mathrm{GAP}_H(x)))), \tag{1}$$

$$W_d(x) = \sigma\left(\left(\sum_{i=1}^{N} \lambda_i W_i\right) \times x\right), \tag{2}$$

where σ is the activation function, N is the number of static convolution kernels. GAP_V represents the vertical direction global average pooling operation, and GAP_H represents the horizontal direction global average pooling operation. We can dynamically adjust the convolution kernel according to the different spatial information of the input, which can help the model better capture spatial variations in the input, enhancing its adaptability to complex scenes and changing environments, thus improving overall performance. Our proposed SAD-DS can achieve significant performance gains with very low computational cost.

B. Multi-Scale Sub-Pixel Lightweight Decoder

To reduce the complexity of the decoder while maintaining accuracy, we propose a multi-scale sub-pixel lightweight decoder for reconstructing high-resolution depth images, as shown in the SUB-Decoder in Fig. 2(a). We employ sub-pixel convolution [16], an upsampling technique that transforms a feature map of size W × H × C into a shape of 2 W × 2H × C/n, where n denotes the upsampling factor. Compared to traditional interpolation-based upsampling methods, sub-pixel convolution can preserve fine image details while reducing the number of feature channels after upsampling. This effectively alleviates the computational burden of subsequent convolution operations, achieving decoder lightweighting.

Each stage in the SUB-Decoder comprises a convolutional layer and a sub-pixel convolutional layer. To further enhance the precision of high-resolution image reconstruction, we adopt a multi-scale image reconstruction method [7], setting the scale to 3 to keep the model lightweight at the same time. The multi-scale prediction head consists of a convolution, a sub-pixel convolution, and a sigmoid activation function. Furthermore, we introduce an additional convolutional layer in the final prediction head to refine the depth map, which is of size H × W and represents the ultimate prediction of the network. Following U-Net [22], we skip and connect the features in the encoder with the features in the decoder to achieve the fusion of high-level semantic information and low-level detail information. SUB-Decoder can still maintain good accuracy when the size is only 0.1 M.

C. Network Training Strategy

1) Self-Supervised Monocular Depth Estimation: In self-supervised monocular depth estimation, the depth estimation network DepthNet and pose estimation network PoseNet are jointly optimized by minimizing the photometric reprojection loss between the target image I_t and the synthesized image $I_{s\to t}$. Given an image sequence, the DepthNet predicts the depth map D_t of the image I_t, while the PoseNet predicts the relative camera pose $T_{t\to s}$ between I_t and I_s, $I_s \in \{I_{t-1}, I_{t+1}\}$. Thus, we can synthesize the target image $I_{s\to t}$:

$$I_{s\to t} = I_s \left\langle K\, T_{t\to s} D_t K^{-1} p_t \right\rangle, \tag{3}$$

where, p_t represents the pixel coordinates of I_t, K is the known camera intrinsic matrix, and $<>$ is the sampling operation. Following [23], we combine L1 loss and structural similarity index measurement (SSIM) [24] loss to formulate our photometric error:

$$\mathcal{L}_p\left(I_{s\to t}, I_t\right) = \alpha \frac{1 - SSIM(I_t, I_{s\to t})}{2} + (1-\alpha)\left|I_t - I_{s\to t}\right|, \tag{4}$$

where $\alpha = 0.85$. To avoid false photometric errors of occluded pixels, we adopt per-pixel minimum reprojection loss [7] :

$$\mathcal{L}_p\left(I_{s\to t}, I_t\right) = \min_{s\in[t-1,t+1]} \mathcal{L}_p\left(I_{s\to t}, I_t\right), \tag{5}$$

In addition, to reduce the influence of moving objects in the scene, we use the auto-masking method [7] to filter out pixels that do not change appearance in adjacent frames, the binary mask is computed as:

$$\mu = \left[\min_{s\in[t-1,t+1]} \mathcal{L}_p(I_s, I_t) > \min_{s\in[t-1,t+1]} \mathcal{L}_p(I_{s\to t}, I_t)\right], \tag{6}$$

where [] is the Iverson Bracket. We employ edge-aware smoothness loss [23] to encourage smoothness except across edges:

$$\mathcal{L}_{\text{smooth}} = \left|\partial_x d_t^*\right| e^{-\left|\partial_x I_t\right|} + \left|\partial_x d_t^*\right| e^{-\left|\partial_y I_t\right|}, \tag{7}$$

where $d_t^* = d_t / \overline{d_t}$ is the mean-normalized inverse depth. Thus, our baseline loss function is defined as

$$\mathcal{L}_s = \sum_{i=1}^{S} \left(\mu \mathcal{L}_p + \lambda \mathcal{L}_{smooth}\right), \tag{8}$$

where μ is 1, λ is $1e^{-3}$ and S is is set to 3 scales.

2) Uncertainty-Aware Self-Teaching Through a Parameter-Shared Encoder: The depth estimation network trained in a self-supervised manner is susceptible to noise interference such as non-Lambertian surfaces and moving objects, which will degrade the performance of the model. In reference [25], it has been demonstrated that uncertainty estimation can capture these noises. Following uncertainty estimation in depth estimation [26] and [27], we employ the uncertainty-aware self-teaching strategy to capture these noises and improve the performance of the model. We employ SAD-Depth, trained in a self-supervised manner, as the teacher network, and an extended version of SAD-Depth with an added decoder for uncertainty prediction as the student network. We train the student network by using the depth predicted by the teacher network as pseudo-labels:

$$\mathcal{L}_{\text{d}} = \frac{1}{N} \sum_{i=1}^{N} \frac{1}{\sigma_d\left(x_i\right)} \left|D_s\left(x_i\right) - D_t\left(x_i\right)\right| + \log \sigma_d\left(x_i\right), \tag{9}$$

where N is the number of pixels, x_i is the pixels in the image, D_t is the predicted depth of the teacher network, D_s is the predicted depth of the student network, and σ_d is the uncertainty of the depth. We take the D_s as the final prediction depth of our method. In this way, we can reduce the reliance of the student network on regions with high uncertainty during training, thereby further improving the model's accuracy.

IV. Experiments

A. Implementation Details

KITTI dataset The KITTI [17] dataset provides a vast collection of street scenes used for autonomous driving and computer vision applications. We split the dataset using Eigen split [31], 39180 images for training, 4424 images for validation, and 697 images for testing. The same intrinsics are used for all images during training. The depth obtained by self-supervised monocular depth estimation has scale ambiguity. Following the method in literature [7], we set the scale as the ratio of the predicted depth median to the true depth median for each image.

Make3D dataset The Make3D dataset [32] contains a variety of outdoor environments, including urban streets, farmland, and rural areas. The model trained on the KITTI dataset will be

TABLE I
QUANTITATIVE RESULTS ON KITTI DATASET

Method	Train	Depth Error (↓)				Depth Accuracy (↑)			Model Size(↓)
		Abs Rel	Sq Rel	RMSE	RMSE log	$\delta < 1.25$	$\delta < 1.25^2$	$\delta < 1.25^3$	Params.
FastDepth [12]	D	0.168	-	5.839	-	0.752	0.927	0.977	3.9M
GuideDepth [14]	D	0.142	-	5.194	-	0.799	0.941	0.982	5.8M
Monodepth2-Res18 [7]	M†	0.132	1.044	5.142	0.210	0.845	0.948	0.977	14.3M
Monodepth2-Res50 [7]	M†	0.131	1.023	5.064	0.206	0.849	0.951	0.979	32.5M
R-MSFM3 [11]	M†	0.128	0.965	5.019	0.207	0.853	0.951	0.977	3.5M
R-MSFM6 [11]	M†	0.126	0.944	4.981	0.204	0.857	0.952	0.978	3.8M
Lite-Mono [10]	M†	0.121	0.876	4.918	0.199	0.859	0.953	0.980	3.1M
SAD-Depth(Ours)	M†	**0.118**	**0.855**	**4.832**	**0.195**	**0.862**	**0.954**	**0.981**	**2.6M**
Zhou [6]	M	0.183	1.595	6.709	0.270	0.734	0.902	0.959	31.6M
DDVO [9]	M	0.151	1.257	5.583	0.228	0.810	0.936	0.974	28.1M
GeoNet [8]	M	0.149	1.060	5.567	0.226	0.796	0.935	0.975	31.6M
Monodepth2-Res18 [7]	M	0.115	0.903	4.863	0.193	0.877	0.959	0.981	14.3M
Monodepth2-Res50 [7]	M	0.110	0.831	4.642	0.187	0.883	0.962	0.982	32.5M
PackNet-SfM [28]	M	0.111	0.785	4.601	0.189	0.878	0.960	0.982	128M
CADepth-Res18 [29]	M	0.110	0.812	4.686	0.187	0.882	0.962	0.983	18.8M
HR-Depth [18]	M	0.109	0.792	4.632	0.185	0.884	0.962	0.983	14.7M
Lite-HR-Depth [18]	M	0.116	0.845	4.841	0.190	0.866	0.957	0.982	3.1M
R-MSFM3 [11]	M	0.114	0.815	4.712	0.193	0.876	0.959	0.981	3.5M
R-MSFM6 [11]	M	0.112	0.806	4.704	0.191	0.878	0.96	0.981	3.8M
MonoFormer [30]	M	0.108	0.806	4.594	0.184	0.884	0.963	0.983	23.9M
Lite-Mono [10]	M	0.107	0.765	**4.561**	0.183	**0.886**	**0.963**	0.983	3.1M
SAD-Depth(Ours)	M	**0.106**	**0.749**	4.591	**0.182**	0.881	0.962	**0.984**	**2.6M**

All input images are resized to 640 × 192. "M": self-supervised learning with pre-training on imagenet. "M†": self-supervised learning without pre-training on imagenet. "D": supervised learning with depth ground truth. The best results are in bold font.

tested on the Make3D dataset which contains 134 test images of outdoor scenes.

Hyperparameters The proposed method is implemented in PyTorch [33]. In our experiments, AdamW [34] is the optimizer and the batch size is set to 12. Posenet uses a pre-trained ResNet18 [35]. Firstly, we perform self-supervised training on the proposed SAD-Depth. For the model trained from scratch, the initial learning rate is 0.0005 and the epoch is set to 50. For the model pre-trained on ImageNet [36], the initial learning rate is set to 0.0001 and the epoch is set to 30. Then, SAD-Depth is trained by an uncertainty-aware self-teaching strategy with the learning rate set to 0.0001 and the epoch set to 25.

Evaluation metrics For depth evaluation, we follow the standard evaluation metrics proposed by Eigen et al. [4] including depth error and depth accuracy. We measure the size of the model using the number of parameters (Params.) and the complexity of the model using the number of floating-point operations (FLOPs). Furthermore, we evaluate the inference speed of the model in the same deployment environment.

B. Depth Estimation Performance

1) KITTI Eigen Split: We conduct our experiments on the KITTI [17] dataset. The final SAD-Depth is the student network without the uncertainty decoder. The experimental results are shown in Table I. Compared with other representative self-supervised monocular depth estimation methods, our proposed SAD-Depth method achieves comparable results while significantly reducing the model parameters. SAD-Depth exhibits superior accuracy compared to the classical Monodepth2-Res18 [7], while having approximately 80 % fewer parameters. In the case of fully convolution neural networks, SAD-Depth outperforms the majority of them. Moreover, SAD-Depth surpasses the recent well-performing CNN and transformer hybrid architecture Lite-Mono [10] on some metrics.

TABLE II
QUALITATIVE RESULTS ON THE MAKE3D DATASET

Method	Abs Rel	Sq Rel	RMSE	RMSE log
Zhou [6]	0.383	5.321	10.470	0.478
DDVO [9]	0.387	4.720	8.090	0.204
Monodepth2 [7]	**0.322**	3.589	7.417	**0.163**
R-MSFM6 [11]	0.334	3.285	7.212	0.169
SAD-Depth	0.325	**3.100**	**7.018**	0.164

All input images are resized to 640 × 192.

We present qualitative results in Fig. 3. It can be observed that our method produces satisfactory results in regions with low texture (traffic sign lines in column 1 and glass windows in column 2) and thin structures (railing in column 3 and tree trunk in column 4). These quantitative and qualitative results demonstrate the superiority of our approach. While maintaining the lightweight design, our model exhibits more consistent depth across the entire scene.

2) Make3D: To verify the generalization ability of our method in different outdoor scenarios, we conducted tests on the Make3D [32] dataset using the model trained on the KITTI dataset. The results are shown in Table II. SAD-Depth exhibits better generalization ability compared to some representative fully convolution neural networks. Fig. 4 shows that SAD-Depth has much sharper edges (columns 1 and 2) as well as weakly textured regions with smooth transitions (column 3). We attribute this to the proposed spatial-aware dynamic depthwise separable convolution module, which can adapt to different scenes.

C. Complexity and Speed Evaluation

The size and inference speed of depth estimation models are crucial for their practical deployment. Table III details the inference speed, floating-point operations, and number of parameters of our model, comparing it with other advanced methods. As

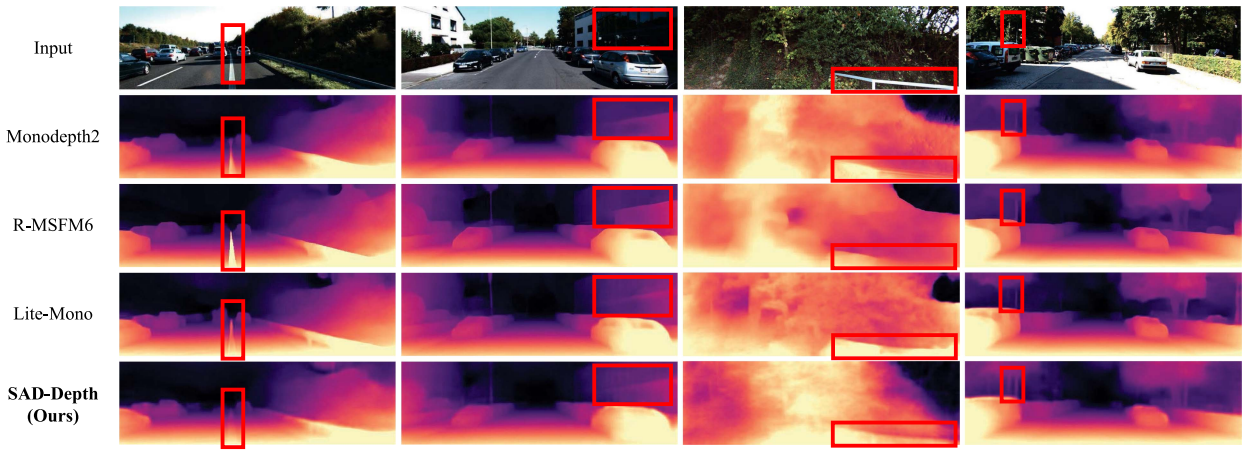


Fig. 3. Qualitative results on KITTI [17] dataset. The first row shows the input RGB image, rows 2–5 display depth maps obtained by different methods. Compared to other methods, our method performs better on weakly textured areas (columns 1 and 2) and thin structures (columns 3 and 4).

TABLE III
MODEL COMPLEXITY AND SPEED EVALUATION

Method	Encoder		Decoder		Full Model		Speed(ms)	
	Params.(M)	FLOPs(G)	Params.(M)	FLOPs(G)	Params.(M)	FLOPs(G)	GPU	CPU
Monodepth2 [7]	11.2	4.5	3.1	3.5	14.3	8	1.49	93.11
R-MSFM3 [11]	0.7	2.4	2.8	14.1	3.5	16.5	2.54	180.43
Lite-Mono [10]	2.9	4.4	0.2	0.7	3.1	5.1	1.84	110.69
SAD-Depth(Ours)	2.5	3.7	0.1	0.3	**2.6**	**4.0**	**1.27**	**82.42**

The input size is 640 × 192. The batch size is 16 when testing reasoning speed.

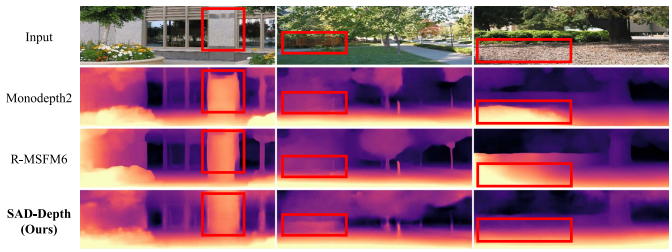


Fig. 4. Qualitative results on Make3D dataset. Compared to other methods, our method has sharper edges (columns 1 and 2) and weakly textured areas with smooth transitions (column 3).

TABLE IV
QUANTITATIVE RESULTS ON "N", THE NUMBER OF CONVOLUTION KERNELS IN SAD-DS MODULES WITHOUT PRE-TRAINING ON IMAGENET

N	Depth Error (↓)				Depth Accuracy (↑)		
	Abs Rel	Sq Rel	RMSE	RMSE log	$\delta <$ 1.25	$\delta <$ 1.25^2	$\delta <$ 1.25^3
2	0.120	**0.824**	4.880	0.197	0.855	0.953	**0.981**
3	**0.118**	0.855	4.832	**0.195**	**0.862**	**0.954**	**0.981**
4	0.120	0.848	**4.810**	**0.195**	0.857	**0.954**	**0.981**
5	0.121	0.849	4.841	0.197	0.856	0.953	**0.981**

TABLE V
QUANTITATIVE RESULTS ON DILATION RATES IN SAD-DS MODULES WITHOUT PRE-TRAINING ON IMAGENET

Dilation Rates	Depth Error (↓)				Depth Accuracy (↑)		
	Abs Rel	Sq Rel	RMSE	RMSE log	$\delta <$ 1.25	$\delta <$ 1.25^2	$\delta <$ 1.25^3
A	**0.118**	**0.855**	4.832	**0.195**	**0.862**	**0.954**	**0.981**
B	0.123	0.873	4.885	0.199	0.853	0.952	0.980
C	0.121	0.861	**4.818**	0.196	0.857	0.953	0.980
D	0.123	0.929	4.884	0.199	0.857	0.952	0.980

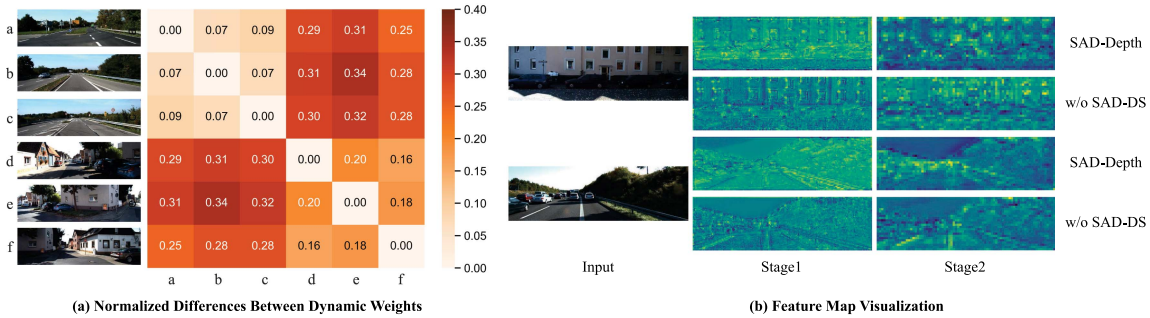
can be seen, our model performs well in terms of both model size and computational complexity. We test the inference speed of the models on both NVIDIA GeForce RTX 3090 and i5-13500H. SAD-Depth achieves significantly faster inference speed on both GPU and CPU processors compared to other models, which makes it suitable for deployment on edge devices. Our method performs on par with the state of the art Lite-Mono [10] in terms of accuracy evaluation. However, the transformer module of Lite-Mono leads to slower inference speed, whereas our model, which employs the SAD-DS module, ensures both accuracy and faster inference speed.

D. Analysis on SAD-DS Modules

1) Analysis of the Context Learned by SAD-DS Modules: To analyze the contextual information learned by SAD-DS modules, we conduct a quantitative analysis of their dynamic weights. Specifically, we utilize the trained SAD-Depth model for inference on images representing highway (a, b, c) and urban street (d, e, f) scenes to acquire their dynamic weights. Subsequently, we calculate the 'normalized differences', which are Euclidean distances between dynamic weights, quantitatively measuring their similarity. A smaller normalized difference indicates a higher similarity, whereas a larger normalized difference suggests a lower similarity. As shown in Fig. 5(a), we observe that the dynamic weights of images from the same scene exhibit smaller normalized differences, while the dynamic lambda weights of images from different scenes show larger normalized differences, aligning with our expectations. This

TABLE VI
ABLATION STUDY ON MODEL ARCHITECTURES WITH PRE-TRAINING ON IMAGENET

Method	Depth Error (↓)				Depth Accuracy (↑)			Model Size(↓)
	Abs Rel	Sq Rel	RMSE	RMSE log	$\delta < 1.25$	$\delta < 1.25^2$	$\delta < 1.25^3$	Params.
FULL	**0.106**	**0.749**	**4.591**	**0.182**	0.881	**0.962**	**0.984**	2.594M
w/o SAD-DS	0.109	0.772	4.599	0.183	0.880	0.961	0.983	**2.557M**
w/o SUB-Decoder	0.107	0.783	4.598	0.183	**0.882**	**0.962**	0.983	2.697M



(a) Normalized Differences Between Dynamic Weights (b) Feature Map Visualization

Fig. 5. Results on the context learned by SAD-DS modules. (a) Normalized differences between dynamic weights: Obtain dynamic weights for the images of highway scenes (a, b, c) and urban street scenes (d, e, f), and then use these dynamic weights to compute the normalized differences between the images. (b) Feature map visualization: The feature maps are obtained from stage 1 and stage 2 of the encoder, showing a comparison between the feature maps with and without the SAD-DS module.

confirms that SAD-DS modules can capture contextual information from various input images and generate representative dynamic weights, which can characterize the spatial information within the images.

In order to provide a more intuitive visualization of the impact of SAD-DS modules on input images, we separately generate the average feature maps of the encoder in the first two stages for both models, one with the SAD-DS module and the other without it. As shown in Fig. 5(b), it can be observed that the model with SAD-DS modules enhances the edge contours in different scenes. The results indicate that SAD-DS modules have excellent adaptive capabilities for various input images, enabling them to capture features from different scenes and enhance their edge contours.

2) Analysis of the Number of Convolution Kernels in SAD-DS Modules: We conduct comparative experiments to investigate the impact of the number of convolutional kernels in the SAD-DS module, denoted as N. These models are trained from scratch, and the results are presented in Table IV. When N = 2, the model performance decreases due to the lower complexity of the convolutional kernels, which cannot fully capture the complex patterns and features in the data. Increasing the value of N enhances the model's complexity and improves its feature extraction capability. However, when N is relatively large, redundancy arises among convolutional kernels, resulting in the model wasting parameters to represent similar information. To strike a balance between computational complexity and accuracy, we set N to 3.

3) Analysis of Dilation Rates in SAD-DS Modules: We analyze the dilation rates in SAD-DS modules of SAD-Depth trained from scratch, referring to the dilation rate settings in Lite-Mono [10], as shown in Table V. By default, "A" divides SAD-DS modules into four groups, using dilation rates of 1, 2, and 3 for the first three groups, and dilation rates of 2, 4, and 6 for the last group. "B" adopts a configuration with dilation rates of 1, 2, and 5 for each group. "C" employs dilation rates of 2, 4, 6 for the third group and 4, 8, 12 for the last group. "D" sets the dilation rates for the last three blocks to 1, 2, and 3. The experimental results indicate that "A" exhibits the best performance. This can be attributed to relatively smaller dilation rates in the early stages of the encoder, which contribute to capturing fine-grained details, while larger dilation rates in the later stages capture a broader range of contextual information. In comparison, "C" exhibits slightly lower accuracy due to excessively large dilation rates in the later stages, leading to the loss of local information. "B" and "D" with smaller dilation rates in the later stages struggle to capture larger-scale features, leading to the omission of extensive contextual information.

E. Ablation Study on Network Architectures

We conduct ablation experiments on the KITTI dataset under the pretraining setting to evaluate the importance of different modules in our method. The results are shown in Table VI.

1) The Benefit of SAD-DS Modules: When all SAD-DS modules in the model are replaced with standard depthwise separable convolutions with the same dilation rates, the accuracy of the model decreases. This demonstrates that our proposed SAD-DS module, which adaptively generates dynamic convolutions for different inputs based on spatial information, enhances the adaptability of the model to complex scenes. This improvement in performance was achieved with only a minor increase in parameter count of 0.037 M.

2) The Benefit of SUB-Decoder: We replaced our designed SUB-Decoder with the bilinear upsampling decoder used in Lite-Mono [10]. It can be observed that the SUB-Decoder, while maintaining good accuracy, has only half the number of parameters compared to bilinear upsampling. This demonstrates

that our designed decoder is not only lightweight but can also reconstruct high-resolution depth images.

V. Conlusion

In this letter, we propose a spatial-aware dynamic lightweight self-supervised monocular depth estimation method. Specifically, we design a spatial-aware dynamic encoder, aiming to overcome the performance limitations of depthwise separable convolution and enhance the feature extraction capability. In the decoder, we utilize sub-pixel upsampling to generate high-quality depth images with relatively low computational resources. Our experimental results on the KITTI dataset demonstrate the superiority of the proposed method and validate the generalization ability of the model on the Make3D dataset. Moreover, SAD-Depth achieves a significant reduction in model complexity and inference speed, enabling efficient and accurate depth estimation in resource-constrained environments. In the future, we plan to further leverage dynamic networks to improve the accuracy of depth estimation models while reducing model size and accelerating inference speed.

References

[1] K. N. McGuire, G. C. d. Croon, C. d. Wagter, K. Tuyls, and H. J. Kappen, "Efficient optical flow and stereo vision for velocity estimation and obstacle avoidance on an autonomous pocket drone," *IEEE Robot. Automat. Lett.*, vol. 2, no. 2, pp. 1070–1076, Apr. 2017.

[2] K. Schmid, T. Tomic, F. Ruess, H. Hirschmüller, and M. Suppa, "Stereo vision based indoor/outdoor navigation for flying robots," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, 2013, pp. 3955–3962.

[3] J. P. C. Valentin et al., "Depth from motion for smartphone AR," *ACM Trans. Graph.*, vol. 37, pp. 1–19, 2018.

[4] D. Eigen, C. Puhrsch, and R. Fergus, "Depth map prediction from a single image using a multi-scale deep network," in *Proc. Annu. Conf. Neural Inf. Process. Syst.*, 2014, vol. 27.

[5] Y. Cao, Z. Wu, and C. Shen, "Estimating depth from monocular images as classification using deep fully convolutional residual networks," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 28, no. 11, pp. 3174–3182, Nov. 2018.

[6] T. Zhou, M. A. Brown, N. Snavely, and D. G. Lowe, "Unsupervised learning of depth and ego-motion from video," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2017, pp. 6612–6619.

[7] C. Godard, O. M. Aodha, and G. J. Brostow, "Digging into self-supervised monocular depth estimation," in *Proc. IEEE/CVF Int. Conf. Comput. Vis.*, 2018, pp. 3827–3837.

[8] Z. Yin and J. Shi, "GeoNet: Unsupervised learning of dense depth, optical flow and camera pose," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2018, pp. 1983–1992.

[9] C. Wang, J. M. Buenaposada, R. Zhu, and S. Lucey, "Learning depth from monocular videos using direct methods," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2018, pp. 2022–2030.

[10] N. Zhang, F. Nex, G. Vosselman, and N. Kerle, "Lite-Mono: A lightweight CNN and transformer architecture for self-supervised monocular depth estimation," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2023, pp. 18537–18546.

[11] Z. Zhou, X. Fan, P. Shi, and Y. Xin, "R-MSFM: Recurrent multi-scale feature modulation for monocular depth estimating," in *Proc. IEEE/CVF Int. Conf. Comput. Vis.*, 2021, pp. 12757–12766.

[12] D. Wofk, F. Ma, T.-J. Yang, S. Karaman, and V. Sze, "Fastdepth: Fast monocular depth estimation on embedded systems," in *Proc. IEEE Int. Conf. Robot. Automat.*, 2019, pp. 6101–6108.

[13] X. Dong, M. A. Garratt, S. G. Anavatti, H. A. Abbass, and J. Dong, "Lightweight monocular depth estimation with an edge guided network," in *Proc. IEEE 17th Int. Conf. Control, Automat., Robot. Vis.*, 2022, pp. 204–210.

[14] M. B. Rudolph, Y. Dawoud, R. Guldenring, L. Nalpantidis, and V. Belagiannis, "Lightweight monocular depth estimation through guided decoding," in *Proc. Int. Conf. Robot. Automat.*, 2022, pp. 2344–2350.

[15] A. G. Howard et al., "MobileNets: Efficient convolutional neural networks for mobile vision applications," 2017, *arXiv:1704.04861*.

[16] W. Shi et al., "Real-time single image and video super-resolution using an efficient sub-pixel convolutional neural network," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2016, pp. 1874–1883.

[17] A. Geiger, P. Lenz, C. Stiller, and R. Urtasun, "Vision meets robotics: The KITTI dataset," *Int. J. Robot. Res.*, vol. 32, pp. 1231–1237, 2013.

[18] X. Lyu et al., "HR-Depth: High resolution self-supervised monocular depth estimation," in *Proc. Conf. Artif. Intell.*, 2021, vol. 35, no. 3, pp. 2294–2301.

[19] X. Zhang, X. Zhou, M. Lin, and J. Sun, "ShuffleNet: An extremely efficient convolutional neural network for mobile devices," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2018, pp. 6848–6856.

[20] S. Mehta, M. Rastegari, A. Caspi, L. G. Shapiro, and H. Hajishirzi, "ESPNet: Efficient spatial pyramid of dilated convolutions for semantic segmentation," in *Proc. Eur. Conf. Comput. Vis.*, 2018, pp. 552-568.

[21] B. Yang, G. Bender, Q. V. Le, and J. Ngiam, "CondConv: Conditionally parameterized convolutions for efficient inference," in *Proc. Annu. Conf. Inf. Process. Syst.*, 2019, vol. 32.

[22] O. Ronneberger, P. Fischer, and T. Brox, "U-Net: Convolutional networks for biomedical image segmentation," in *Proc. 18th Int. Conf. Med. Image Comput. Comput.-Assisted Intervention*, 2015, pp. 234–241.

[23] C. Godard, O. M. Aodha, and G. J. Brostow, "Unsupervised monocular depth estimation with left-right consistency," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2017, pp. 6602–6611.

[24] Z. Wang, A. C. Bovik, H. R. Sheikh, and E. P. Simoncelli, "Image quality assessment: From error visibility to structural similarity," *IEEE Trans. Image Process.*, vol. 13, no. 4, pp. 600–612, Apr. 2004.

[25] A. Kendall and Y. Gal, "What uncertainties do we need in Bayesian deep learning for computer vision?," in *Proc. 31st Int. Conf. Neural Inf. Process. Syst.*, 2017, pp. 5580-5590.

[26] M. Poggi, F. Aleotti, F. Tosi, and S. Mattoccia, "On the uncertainty of self-supervised monocular depth estimation," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2020, pp. 3224–3234.

[27] X. Nie, D. X. Shi, R. Li, Z. Liu, and X. Chen, "Uncertainty-aware self-improving framework for depth estimation," *IEEE Robot. Automat. Lett.*, vol. 7, no. 1, pp. 41–48, Jan. 2022.

[28] V. C. Guizilini, R. Ambrus, S. Pillai, and A. Gaidon, "3D packing for self-supervised monocular depth estimation," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2020, pp. 2482–2491.

[29] J. Yan, H. Zhao, P. Bu, and Y. Jin, "Channel-wise attention-based network for self-supervised monocular depth estimation," in *Proc. IEEE Int. Conf. 3D Vis.*, 2021, pp. 464–473.

[30] J.-H. Bae, S. Moon, and S. Im, "Monoformer: Towards generalization of self-supervised monocular depth estimation with transformers," 2022, *arXiv:2205.11083*.

[31] 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.*, 2014, pp. 2650–2658.

[32] A. Saxena, M. Sun, and A. Ng, "Make3D: Learning 3D scene structure from a single still image," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 31, no. 5, pp. 824–840, May 2009.

[33] A. Paszke et al., "Automatic differentiation in PyTorch," *NeurIPS Workshop*, 2017.

[34] I. Loshchilov and F. Hutter, "Decoupled weight decay regularization," in *Proc. Int. Conf. Learn. Representations*, 2017.

[35] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2016, pp. 770–778.

[36] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, "ImageNet: A large-scale hierarchical image database," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2009, pp. 248–255.