Lite-Mono: A Lightweight CNN and Transformer Architecture for Self-Supervised Monocular Depth Estimation

Ning Zhang*   Francesco Nex   George Vosselman   Norman Kerle

University of Twente

{n.zhang, f.nex, george.vosselman, n.kerle}@utwente.nl

Abstract

Self-supervised monocular depth estimation that does not require ground truth for training has attracted attention in recent years. It is of high interest to design lightweight but effective models so that they can be deployed on edge devices. Many existing architectures benefit from using heavier backbones at the expense of model sizes. This paper achieves comparable results with a lightweight architecture. Specifically, the efficient combination of CNNs and Transformers is investigated, and a hybrid architecture called Lite-Mono is presented. A Consecutive Dilated Convolutions (CDC) module and a Local-Global Features Interaction (LGFI) module are proposed. The former is used to extract rich multi-scale local features, and the latter takes advantage of the self-attention mechanism to encode long-range global information into the features. Experiments demonstrate that Lite-Mono outperforms Monodepth2 by a large margin in accuracy, with about 80% fewer trainable parameters. Our codes and models are available at https://github.com/noahzn/Lite-Mono.

1. Introduction

Many applications in the field of robotics, autonomous driving, and augmented reality rely on depth maps, which represent the 3D geometry of a scene. Since depth sensors increase costs, research on inferring depth maps using Convolutional Neural Networks (CNNs) from images emerged. With the annotated depth one can train a regression CNN to predict the depth value of each pixel on a single image [10, 11, 22]. Lacking large-scale accurate dense ground-truth depth for supervised learning, self-supervised methods that seek supervisory signals from stereo-pairs of frames or monocular videos are favorable and have made great progress in recent years. These methods regard the depth estimation task as a novel view synthesis problem and minimize an image reconstruction loss [5,14,15,41,45].
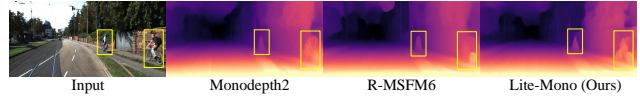
*Corresponding Author

Figure 1. The proposed Lite-Mono has fewer parameters than Monodepth2 [15] and R-MSFM [46], but generates more accurate depth maps.

The camera motion is known when using stereo-pairs of images, so a single depth estimation network is adopted to predict depth. But if only using monocular videos for training an additional pose network is needed to estimate the motion of the camera. Despite this, self-supervised methods that only require monocular videos are preferred, as collecting stereo data needs complicated configurations and data processing. Therefore, this paper also focuses on monocular video training.

In addition to increasing the accuracy of monocular training by introducing improved loss functions [15] and semantic information [5, 21] to mitigate the occlusion and moving objects problems, many works focused on designing more effective CNN architectures [17, 33, 39, 41, 46]. However, the convolution operation in CNNs has a local receptive field, which cannot capture long-range global information. To achieve better results a CNN-based model can use a deeper backbone or a more complicated architecture [15, 28, 44], which also results in a larger model size. The recently introduced Vision Transformer (ViT) [8] is able to model global contexts, and some recent works apply it to monocular depth estimation architectures [3, 35] to obtain better results. However, the expensive calculation of the Multi-Head Self-Attention (MHSA) module in a Transformer hinders the design of lightweight and fast inference models, compared with CNN models [35].

This paper pursues a lightweight and efficient self-supervised monocular depth estimation model with a hybrid CNN and Transformer architecture. In each stage of the proposed encoder a Consecutive Dilated Convolutions (CDC) module is adopted to capture enhanced multi-scale local features. Then, a Local-Global Features Interaction (LGFI) module is used to calculate the MHSA and encode

global contexts into the features. To reduce the computational complexity the ==cross-covariance attention== [1] is calculated in the channel dimension instead of the spatial dimension. The contributions of this paper can be summarized in three aspects.

- A new lightweight architecture, dubbed Lite-Mono, for self-supervised monocular depth estimation, is proposed. Its effectiveness with regard to the model size and FLOPs is demonstrated.

- The proposed architecture shows superior accuracy on the KITTI [13] dataset compared with competitive larger models. It achieves state-of-the-art with the least trainable parameters. The model's generalization ability is further validated on the Make3D [32] dataset. Additional ablation experiments are conducted to verify the effectiveness of different design choices.

- The inference time of the proposed method is tested on an NVIDIA TITAN Xp and a Jetson Xavier platform, which demonstrates its good trade-off between model complexity and inference speed.

The remainder of the paper is organized as follows. Section 2 reviews some related research work. Section 3 illustrates the proposed method in detail. Section 4 elaborates on the experimental results and discussion. Section 5 concludes the paper.

2. Related work

2.1. Monocular depth estimation using deep learning

Single image depth estimation is an ill-posed problem, because a 2D image may correspond to many 3D scenes at different scales. Methods using deep learning can be roughly divided into two categories.

Supervised depth estimation. Using ground-truth depth maps as supervision, a supervised deep learning network is able to extract features from input images and learn the relationship between depth and RGB values. Eigen *et al.* [10] first used deep networks to estimate depth maps from single images. They designed a multi-scale network to combine global coarse depth maps and local fine depth maps. Subsequent works introduced some post-processing techniques, such as Conditional Random Fields (CRF), to improve the accuracy [24,25,38]. Laina *et al.* [22] proposed to use a new up-sampling module and the reverse Huber loss to improve the training. Fu *et al.* [11] adopted a multi-scale network, and treated the depth estimation as an ordinal regression task. Their method achieved higher accuracy and faster convergence.

Self-supervised depth estimation. Considering that large-scale annotated datasets are not always available, self-supervised depth estimation methods that do not require

ground truth for training have attracted some attention. Garg *et al.* [12] regarded the depth estimation as a novel view synthesis problem, and proposed to minimize a photometric loss between an input left image and the synthesized right image. Their method was self-supervised, as the supervisory signal came from the input stereo pairs. Godard *et al.* [14] extended this work and achieved higher accuracy by introducing a left-right disparity consistency loss. Apart from using stereo pairs the supervisory signal can also come from monocular video frames. Zhou *et al.* [45] trained a separate multi-view pose network to estimate the pose between two sequential frames. To improve the robustness when dealing with occlusion and moving objects they also used an explainability prediction network to ignore target pixels that violate view synthesis assumptions. To model dynamic scenes other works introduced multi-task learning, such as optical flow estimation [41] and semantic segmentation [5,20], or introduced additional constraints, such as uncertainty estimation [30,40]. Godard *et al.* [15] found that without introducing extra learning tasks they could achieve competitive results by simply improving the loss functions. They proposed Monodepth2, which used a minimum reprojection loss to mitigate occlusion problems, and an auto-masking loss to filter out moving objects that have the same velocity as the camera. This work is also based on their self-supervised training strategy.

2.2. Advanced architectures for depth estimation

Network architectures also play an important role in achieving good results in monocular depth estimation. By replacing the network architecture from the VGG model [33] with a ResNet [17] Yin *et al.* [41] achieved better results. Yan *et al.* [39] used a channel-wise attention module to capture long-range multi-level information and enhance local features. Zhou *et al.* [44] also used an attention module to obtain a better feature fusion. Zhao *et al.* [46] proposed a small architecture that used a feature modulation module to learn multi-scale features, and demonstrated the method's superiority. To reduce model parameters they only used the first three stages of ResNet18 [17] as the backbone. With the rise of vision transformer (ViT) [8] recent work applied it to various computer vision tasks [4,16,29,31,34], and achieved promising results. However, research incorporating Transformers in depth estimation architectures is still limited. Varma *et al.* [35] adopted the Dense Prediction Transformer [31] for self-supervised monocular depth estimation, and added another prediction head to estimate the camera's intrinsic. Bae *et al.* [3] proposed a hybrid architecture of CNN and Transformer that enhanced CNN features by Transformers. However, due to the high computational complexity of Multi-Head Self-Attention (MHSA) in a ViT, the above-mentioned Transformer-based methods have more
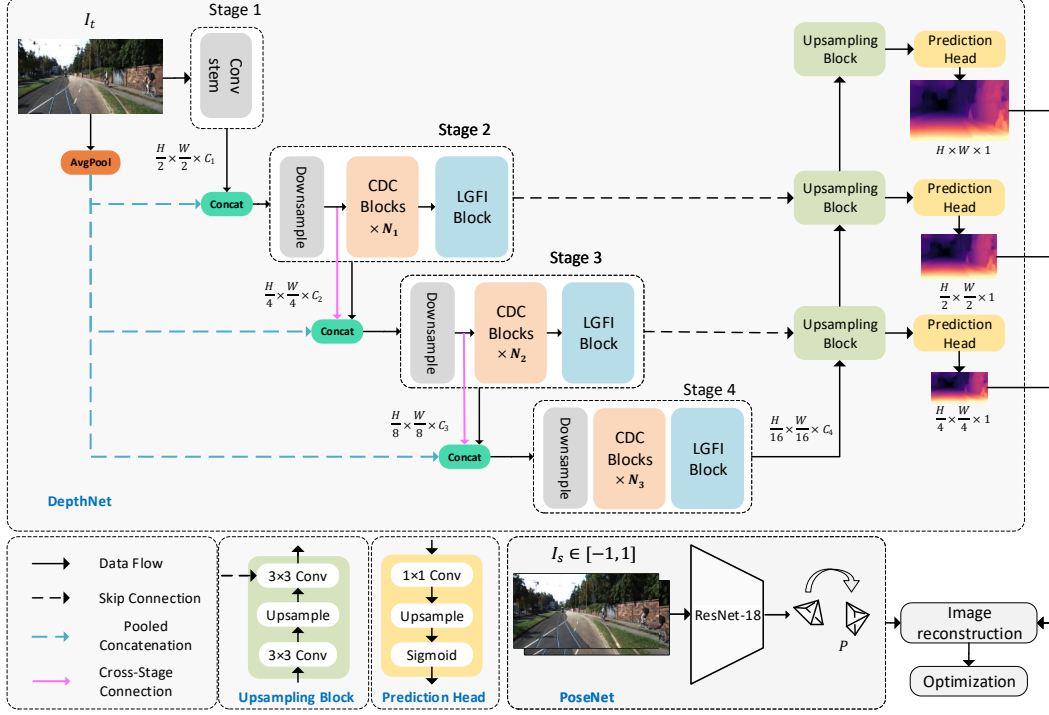
Figure 2. **Overview of the proposed Lite-Mono.** Lite-Mono has an encoder-decoder DepthNet for depth prediction, and a commonly used PoseNet [15, 46] to estimate poses between adjacent monocular frames. The encoder of the DepthNet consists of four stages, and it uses Consecutive Dilated Convolutions (CDC) modules and Local-Global Features Interaction (LGFI) modules to extract rich hierarchical features. The details of these modules are shown in Figure 3.

trainable parameters and have a large speed gap compared with methods only using CNNs [35]. MonoViT [43] uses MPViT [23] as its encoder and has achieved state-of-the-art accuracy. Nevertheless, the use of multiple parallel blocks in MonoViT slows down its speed.

3. The proposed framework: Lite-Mono

3.1. Design motivation and choices

Several papers demonstrated that a good encoder can extract more effective features, thus improving the final result [15, 17, 44]. This paper focuses on designing a lightweight encoder that can encode effective features from the input images. Figure 2 shows the proposed architecture. It consists of an encoder-decoder DepthNet (Section 3.2) and a PoseNet (Section 3.3). The DepthNet estimates multi-scale inverse depth maps of the input image, and the PoseNet estimates the camera motion between two adjacent frames. Then, a reconstructed target image is generated, and the loss is computed to optimize the model (Section 3.4).

Enhanced local features. Using shallow instead of deeper networks can effectively reduce the size of a model. As mentioned shallow CNNs have very limited receptive fields, while using dilated convolution [42] is helpful to en-

large receptive fields. By stacking the proposed Consecutive Dilated Convolutions (CDC) the network is able to "observe" the input at a larger area, while not introducing extra training parameters.

Low-computation global information. The enhanced local features are not enough to learn a global representation of the input without the help of Transformers to model long-range information. The MHSA module in the original Transformer [8] has a linear computational complexity to the input dimension, hence it limits the design of lightweight models. Instead of computing the attention across the spatial dimension the proposed Local-Global Features Interaction (LGFI) module adopts the cross-covariance attention [1] to compute the attention along the feature channels. Comparing with the original self-attention [8] it reduces the memory complexity from $\mathcal{O}(hN^2+Nd)$ to $\mathcal{O}(d^2/h+Nd)$, and reduces the time complexity from $\mathcal{O}(N^2d)$ to $\mathcal{O}(Nd^2/h)$, where h is the number of attention heads. The proposed architecture is described in detail below.

3.2. DepthNet

Depth encoder. The proposed Lite-Mono aggregates multi-scale features across four stages. The input image

Output Size	Layers	Lite-Mono-tiny	Lite-Mono-small	Lite-Mono	Lite-Mono-8M
640×192	Input				
320×96	Conv Stem	$3 \times 3, 32, stride = 2$ $[3 \times 3, 32] \times 2$	$3 \times 3, 48, stride = 2$ $[3 \times 3, 48] \times 2$	$3 \times 3, 48, stride = 2$ $[3 \times 3, 48] \times 2$	$3 \times 3, 64, stride = 2$ $[3 \times 3, 64] \times 2$
160×48 Stage 1	Downsampling CDC blocks LGFI block	$3 \times 3, 32, stride = 2$ $[3 \times 3, 32] \times 3$ $dilation = 1, 2, 3$	$3 \times 3, 48, stride = 2$ $[3 \times 3, 48] \times 3$ $dilation = 1, 2, 3$	$3 \times 3, 48, stride = 2$ $[3 \times 3, 48] \times 3$ $dilation = 1, 2, 3$	$3 \times 3, 64, stride = 2$ $[3 \times 3, 64] \times 3$ $dilation = 1, 2, 3$
80×24 Stage 2	Downsampling CDC blocks LGFI block	$3 \times 3, 64, stride = 2$ $[3 \times 3, 64] \times 3$ $dilation = 1, 2, 3$	$3 \times 3, 80, stride = 2$ $[3 \times 3, 80] \times 3$ $dilation = 1, 2, 3$	$3 \times 3, 80, stride = 2$ $[3 \times 3, 80] \times 3$ $dilation = 1, 2, 3$	$3 \times 3, 128, stride = 2$ $[3 \times 3, 128] \times 3$ $dilation = 1, 2, 3$
40×12 Stage 3	Downsampling CDC blocks LGFI block	$3 \times 3, 128, stride = 2$ $[3 \times 3, 128] \times 6$ $dilation = 1, 2, 3, 2, 4, 6$	$3 \times 3, 128, stride = 2$ $[3 \times 3, 128] \times 6$ $dilation = 1, 2, 3, 2, 4, 6$	$3 \times 3, 128, stride = 2$ $[3 \times 3, 128] \times 9$ $dilation = [1, 2, 3] \times 2, 2, 4, 6$	$3 \times 3, 224, stride = 2$ $[3 \times 3, 224] \times 9$ $dilation = [1, 2, 3] \times 2, 2, 4, 6$
#Params. (M)		2.0	2.3	2.9	8.1

Table 1. **Four variants of the proposed depth encoder.** $[3 \times 3, C] \times N$ means that a CDC block uses the 3×3 kernel size to output C channels, and repeats for N times. The dilation rate used in each CDC block is also listed.
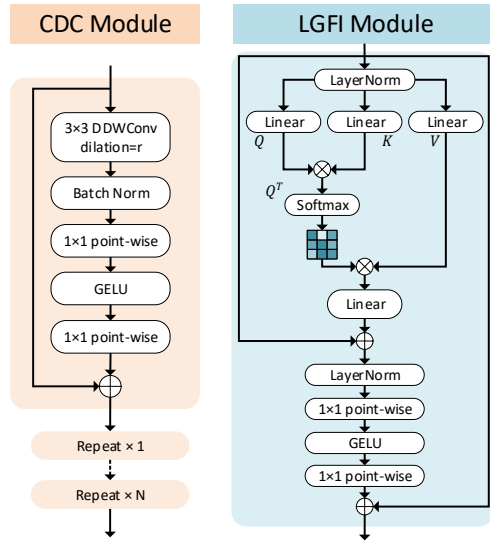

Figure 3. **Structures of the proposed Consecutive Dilated Convolutions (CDC) module and Local-Global Features Interaction (LGFI) module.** In each stage the CDC module with different dilation rate is repeated for N times.

with size $H \times W \times 3$ is first fed into a convolution stem, where the image is down-sampled by a 3×3 convolution. Following two additional 3×3 convolutions with $stride = 1$ for local feature extraction, the feature maps of size $\frac{H}{2} \times \frac{W}{2} \times C_1$ are obtained. In the second stage the features are concatenated with the pooled three-channel input image, and another 3×3 convolution with $stride = 2$ is adopted to down-sample the feature maps, resulting in feature maps with size $\frac{H}{4} \times \frac{W}{4} \times C_2$. Concatenating features with the average-pooled input image in a down-sampling layer can reduce the spatial information loss caused by the reduction of feature size, which is inspired by ESP-Netv2 [3]. Then, the proposed Consecutive Dilated Convolutions (CDC) module and the Local-Global Features Interaction (LGFI) module learn rich hierarchical feature rep-

resentations. The down-sampling layers in the second and the third stage also receive the concatenated features output by the previous down-sampling layer. This design is similar to the residual connection proposed in ResNet [17], and is able to model better cross-stage correlation. Similarly, the output feature maps are further fed into the third and the fourth stages, and output features of dimension $\frac{H}{8} \times \frac{W}{8} \times C_3$ and $\frac{H}{16} \times \frac{W}{16} \times C_4$.

Consecutive Dilated Convolutions (CDC). The proposed CDC module utilizes dilated convolutions to extract multi-scale local features. Different from using a parallel dilated convolution module only in the last layer of the network [6] we insert several consecutive dilated convolutions with different dilation rates into each stage for adequate multi-scale contexts aggregation.

Given a two-dimensional signal $x[i]$ the output $y[i]$ of a 2D dilated convolution can be defined as:

$$y[i] = \sum_{k=1}^{K} x[i + r \cdot k] w[k], \quad (1)$$

where $w[k]$ is a filter with length K, and r denotes the dilation rate used to convolve the input $x[i]$. In a standard non-dilated convolution $r = 1$. By using a dilated convolution the network can keep the size of the output feature map fixed while achieving a larger receptive field. Considering an input feature X with dimension $H \times W \times C$ our CDC module outputs $\hat{X}$ as follows:

$$\hat{X} = X + Linear_G(Linear(BN(DDWConv_r(X)))), \quad (2)$$

where $Linear_G$ denotes a point-wise convolution operation, followed by the $GELU$ [18] activation. BN is a batch normalization layer, and $DDWConv_r(\cdot)$ is a 3×3 depth-wise dilated convolution with dilation rate r.

Local-Global Features Interaction (LGFI). Given an input feature map X with dimension $H \times W \times C$ it is linearly projected to the same dimensional 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 [1] is used to enhance the input X:

$$\widetilde{X} = Attention(Q, K, V) + X, \qquad (3)$$

where $Attention(Q, K, V) = V \cdot Softmax(Q^T \cdot K)$. Then, the non-linearity of the features can be increased:

$$\hat{X} = X + Linear_G(Linear(LN(\widetilde{X}))), \qquad (4)$$

where LN is a layer normalization [2] operation. According to different channel numbers, CDC blocks, and dilation rates, four variants of the depth encoder are designed. Table 1 shows more details.

Depth decoder. Different from using a complicated up-sampling method [46] or introducing additional attention modules [3] Lite-Mono uses a depth decoder adapted from [15]. As shown in Figure 2 it increases the spatial dimension using bi-linear up-sampling, and uses convolutional layers to concatenate features from three stages of the encoder. Each up-sampling block follows a prediction head to output the inverse depth map at full, $\frac{1}{2}$, and $\frac{1}{4}$ resolution, respectively.

3.3. PoseNet

Following [15, 46] this paper uses the same PoseNet for pose estimation. To be specific, a pre-trained ResNet18 is used as the pose encoder, and it receives a pair of color images as input. A pose decoder with four convolutional layers is used to estimate the corresponding 6-DoF relative pose between adjacent images.

3.4. Self-supervised learning

Different from the supervised training that utilizes ground truth of depth this work treats depth estimation as the task of image reconstruction. Similar to [45] the learning objective is modeled to minimize an image reconstruction loss $\mathcal{L}_r$ between a target image I_t and a synthesized target image $\hat{I}_t$, and an edge-aware smoothness loss $\mathcal{L}_{smooth}$ constrained on the predicted depth map D_t.

Image reconstruction loss. The photometric reprojection loss is defined as:

$$\mathcal{L}_p(\hat{I}_t, I_t) = \mathcal{L}_p(\mathcal{F}(I_s, P, D_t, K), I_t), \qquad (5)$$

where $\hat{I}_t$ can be obtained by a function $\mathcal{F}$ in terms of the source image I_s, the estimated pose P, the predicted depth D_t, and the camera's intrinsics K. As introduced in [45] $\mathcal{L}_p$ is computed by a sum of the pixel-wise similarity $SSIM$ (Structural Similarity Index [37]) and the $L1$ loss between $\hat{I}_t$ and I_t:

$$\mathcal{L}_p(\hat{I}_t, I_t) = \alpha \frac{1 - SSIM(\hat{I}_t, I_t)}{2} + (1-\alpha)\|\hat{I}_t - I_t\|, \quad (6)$$

where α is set to 0.85 empirically [15]. In addition, to deal with out-of-view pixels and occluded objects in a source image the minimum photometric loss [15] is computed:

$$\mathcal{L}_p(I_s, I_t) = \min_{I_s \in [-1,1]} \mathcal{L}_p(\hat{I}_t, I_t), \qquad (7)$$

where I_s can be either the previous or the next frame with respect to the target image. Another binary mask [15] is used to remove moving pixels:

$$\mu = \min_{I_s \in [-1,1]} \mathcal{L}_p(I_s, I_t) > \min_{I_s \in [-1,1]} \mathcal{L}_p(\hat{I}_t, I_t). \qquad (8)$$

Therefore, the image reconstruction loss is defined as:

$$\mathcal{L}_r(\hat{I}_t, I_t) = \mu \cdot \mathcal{L}_p(I_s, I_t), \qquad (9)$$

Edge-aware smoothness loss. To smooth the generated inverse depth maps an edge-aware smoothness loss is calculated, followed by [15, 44]:

$$\mathcal{L}_{smooth} = |\partial_x d_t^*| e^{-|\partial_x I_t|} + |\partial_x d_t^*| e^{-|\partial_y I_t|}, \qquad (10)$$

where $d_t^* = d_t/\hat{d}_t$ denotes the mean-normalized inverse depth. The total loss can be expressed as:

$$\mathcal{L} = \frac{1}{3} \sum_{s \in \{1, \frac{1}{2}, \frac{1}{4}\}} (\mathcal{L}_r + \lambda \mathcal{L}_{smooth}), \qquad (11)$$

where s is the different scale output by the depth decoder. λ is set to $1e^{-3}$ as in [15].

4. Experiments

This section evaluates the proposed framework and demonstrates the superiority of Lite-Mono.

4.1. Datasets

KITTI. The KITTI [13] dataset contains 61 stereo road scenes for research in autonomous driving and robotics, and it was collected by multiple sensors, including camera, 3D Lidar, GPU/IMU, etc. To train and evaluate the proposed method the Eigen split [9] is used, which has a total of 39,180 monocular triplets for training, 4,424 for evaluation, and 697 for testing. The self-supervised training is based on the known camera intrinsics K, as indicated in Eq. 5. By averaging all the focal lengths of images across the KITTI dataset this paper uses the same intrinsics for all images during training [15]. In the evaluation the predicted depth is restricted in the range of $[0, 80]$m, as is common practice.

Make3D. To evaluate the generalization ability of the proposed method it is further tested on the Make3D [32] dataset, which contains 134 test images of outdoor scenes. The model trained on the KITTI dataset is loaded and inferred directly on these test images.

Method	Year	Data	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.
GeoNet [41]	2018	M	0.149	1.060	5.567	0.226	0.796	0.935	0.975	31.6M
DDVO [36]	2018	M	0.151	1.257	5.583	0.228	0.810	0.936	0.974	28.1M
Monodepth2-Res18 [15]	2019	M	0.115	0.903	4.863	0.193	0.877	0.959	0.981	14.3M
Monodepth2-Res50 [15]	2019	M	0.110	0.831	4.642	0.187	0.883	_0.962_	_0.982_	32.5M
SGDepth [21]	2020	M+Se	0.113	0.835	4.693	0.191	0.879	0.961	0.981	16.3M
Johnston _et al._ [19]	2020	M	0.111	0.941	4.817	0.189	_0.885_	0.961	0.981	14.3M+
CADepth-Res18 [39]	2021	M	0.110	0.812	4.686	0.187	0.882	_0.962_	**0.983**	18.8M
HR-Depth [28]	2021	M	0.109	_0.792_	4.632	0.185	0.884	_0.962_	**0.983**	14.7M
Lite-HR-Depth [28]	2021	M	0.116	0.845	4.841	0.190	0.866	0.957	_0.982_	3.1M
R-MSFM3 [46]	2021	M	0.114	0.815	4.712	0.193	0.876	0.959	0.981	3.5M
R-MSFM6 [46]	2021	M	0.112	0.806	4.704	0.191	0.878	0.960	0.981	3.8M
MonoFormer [3]	2022	M	_0.108_	0.806	_4.594_	_0.184_	0.884	**0.963**	**0.983**	23.9M+
Lite-Mono-tiny (Ours)	2023	M	0.110	0.837	4.710	0.187	0.880	0.960	_0.982_	**2.2M**
Lite-Mono-small (Ours)	2023	M	0.110	0.802	4.671	0.186	0.879	0.961	_0.982_	2.5M
Lite-Mono (Ours)	2023	M	**0.107**	**0.765**	**4.561**	**0.183**	**0.886**	**0.963**	**0.983**	3.1M
Monodepth2-Res18 [15]	2019	M†	0.132	1.044	5.142	0.210	0.845	0.948	0.977	14.3M
Monodepth2-Res50 [15]	2019	M†	0.131	1.023	5.064	0.206	0.849	0.951	_0.979_	32.5M
R-MSFM3 [46]	2021	M†	0.128	0.965	5.019	0.207	0.853	0.951	0.977	3.5M
R-MSFM6 [46]	2021	M†	0.126	0.944	4.981	0.204	_0.857_	_0.952_	0.978	3.8M
Lite-Mono-tiny (Ours)	2023	M†	0.125	0.935	4.986	0.204	0.853	0.950	0.978	**2.2M**
Lite-Mono-small (Ours)	2023	M†	_0.123_	_0.919_	_4.926_	_0.202_	**0.859**	0.951	0.977	_2.5M_
Lite-Mono (Ours)	2023	M†	**0.121**	**0.876**	**4.918**	**0.199**	**0.859**	**0.953**	**0.980**	3.1M
Monodepth2-Res18 [15]	2019	M*	0.115	0.882	4.701	0.190	0.879	0.961	0.982	14.3M
R-MSFM3 [46]	2021	M*	0.112	0.773	4.581	0.189	0.879	0.960	0.982	3.5M
R-MSFM6 [46]	2021	M*	0.108	0.748	4.470	0.185	0.889	0.963	0.982	3.8M
HR-Depth [28]	2021	M*	0.106	0.755	4.472	0.181	0.892	_0.966_	**0.984**	14.7M
Lite-Mono-tiny (Ours)	2023	M*	0.104	0.764	4.487	0.180	0.892	0.964	_0.983_	**2.2M**
Lite-Mono-small (Ours)	2023	M*	0.103	0.757	4.449	0.180	0.894	0.964	_0.983_	_2.5M_
Lite-Mono (Ours)	2023	M*	_0.102_	0.746	_4.444_	_0.179_	0.896	0.965	_0.983_	3.1M
Lite-Mono-8M (Ours)	2023	M*	**0.097**	**0.710**	**4.309**	**0.174**	**0.905**	**0.967**	**0.984**	8.7M
MonoViT-tiny [43]	2022	M	0.102	0.733	4.459	**0.177**	0.895	**0.965**	**0.984**	10.3M
Lite-Mono-8M (Ours)	2023	M	**0.101**	**0.729**	**4.454**	0.178	**0.897**	**0.965**	0.983	**8.7M**

Table 2. **Comparison of Lite-Mono with some recent representative methods on the KITTI benchmark using the Eigen split [9].** All input images are resized to 640×192 unless otherwise specified. The best and the second best results are highlighted in **bold** and underlined, respectively. ”M”: KITTI monocular videos, ”M+Se”: monocular videos + semantic segmentation, ”M*”: input resolution 1024×320, ”M†”: without pre-training on ImageNet [7].
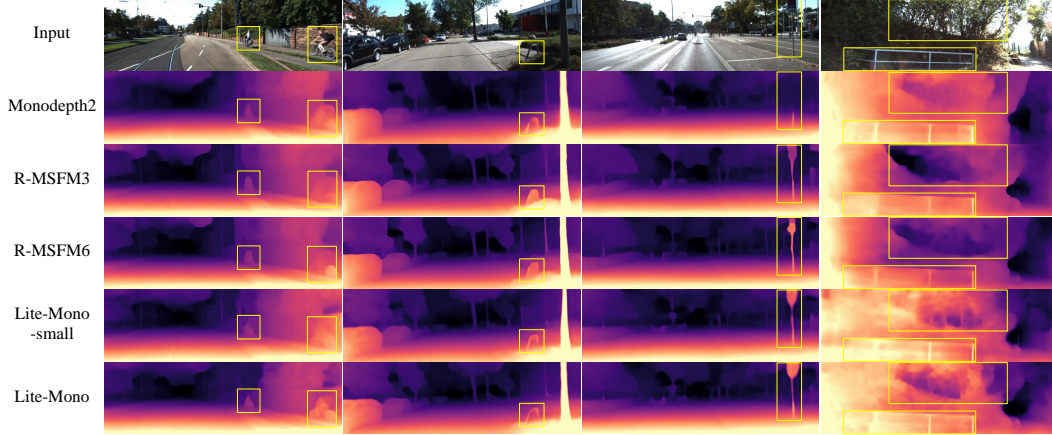

Figure 4. **Qualitative results on KITTI.** Here are some depth maps generated by Monodepth2 [15], R-MSFM3 [46], R-MSFM6 [46], Lite-Mono-small (ours), and Lite-Mono (ours), respectively. Monodepth2 and R-MSFM have limited receptive fields, so they yield some inaccurate depth predictions. Instead, our models can generate better results.

4.2. Implementation details

Hyperparameters. The proposed method is implemented in PyTorch and trained on a single NVIDIA TITAN Xp with a batch size of 12. AdamW [27] is the optimizer, and the weight decay is set to $1e^{-2}$. Drop-path is used in the CDC and LGFI modules to mitigate overfitting. For models trained from scratch an initial learning rate of $5e^{-4}$ with a cosine learning rate schedule [26] is adopted, and the training epoch is set to 35. It is found that pre-training on Ima-

geNet [7] makes the network converge fast, so the network is trained for 30 epochs when using pre-trained weights, and the initial learning rate is set to $1e^{-4}$. A monocular training for 35 epochs takes about 15 hours.

Data augmentation. Data augmentation is adopted as a preprocessing step to improve the robustness of the training. To be specific, the following augmentations are performed with a 50% chance: horizontal flips, brightness adjustment (± 0.2), saturation adjustment (± 0.2), contrast adjustment (± 0.2), and hue jitter (± 0.1). These adjustments are applied in a random order, and the same augmentation method is also used by [15, 28, 46].

Evaluation metrics Accuracy is reported in terms of seven commonly used metrics proposed in [10], which are Abs Rel, Sq Rel, RMSE, RMSE log, $\delta < 1.25$, $\delta < 1.25^2$, and $\delta < 1.25^3$.

4.3. KITTI results

The proposed framework is compared with other representative methods with model sizes less than 35M, and the results are shown in Table 2. Lite-Mono beats all methods except MonoViT-tiny and is the smallest model (3.1M). Specifically, Lite-Mono greatly exceeds Monodepth2 [15] with a ResNet18 [17] backbone, but the model size is only about one-fifth of this model. It also outperforms the ResNet50 version of Monodepth2, which is the largest model (32.5M) in this table. Besides, Lite-Mono surpasses the recent well-designed small model R-MSFM [46]. Compared with the new MonoFormer [3] with a ResNet50 backbone the proposed Lite-Mono outperforms it in all metrics. Our other two smaller models also achieve satisfactory results, considering that they have fewer trainable parameters. In the last two rows of the table, the proposed Lite-Mono-8M also performs better than MonoViT-tiny, the smallest model of MonoViT [43], with fewer parameters. Figure 4 shows that Lite-Mono achieves satisfactory results, even on challenging images where moving objects are close to the camera (column 1).

4.4. Make3D results

The proposed method is evaluated on the Make3D dataset to show its generalization ability in different outdoor scenes. The model trained on KITTI is directly inferred without any fine-tuning. Table 3 shows the comparison of Lite-Mono with the other three methods, and Lite-Mono performs the best. Figure 5 shows some qualitative results. Owing to the proposed feature extraction modules Lite-Mono is able to model both local and global contexts, and perceives objects with different sizes.

4.5. Complexity and speed evaluation

The proposed models' parameters, FLOPs (floating point of operations), and inference time are evaluated on an

Method	Abs Rel	Sq Rel	RMSE	RMSE log
DDVO [36]	0.387	4.720	8.090	0.204
Monodepth2 [15]	0.322	3.589	7.417	0.163
R-MSFM6 [46]	0.334	3.285	7.212	0.169
Lite-Mono (Ours)	**0.305**	**3.060**	**6.981**	**0.158**

Table 3. **Comparison of the proposed Lite-Mono to some other methods on the Make3D [32] dataset.** All models are trained on KITTI [13] with an image resolution of 640×192.
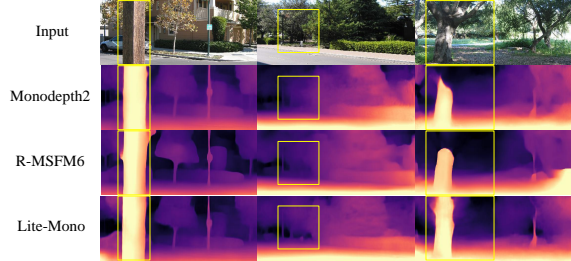

Figure 5. **Qualitative results on the Make3D dataset.** Lite-Mono is compared to Monodepth2 [15] and R-MSFM [46]. Lite-Mono can perceive different sizes of objects.

NVIDIA TITAN Xp and a Jetson Xavier and are compared with Monodepth2 [15], R-MSFM [46], and MonoViT-tiny [43]. Table 4 shows that the proposed design has a good balance between model size and speed. Notice that Lite-Mono-tiny outperforms Monodepth2 both in speed and accuracy (Table 2). Although R-MSFM [46] is a lightweight model it is slow. The latest MonoViT-tiny [43] runs the slowest due to its parallel blocks and multiple layers of self-attention. Our models also infer quickly on the Jetson Xavier, which allows them to be used on edge devices.

4.6. Ablation study on model architectures

To further demonstrate the effectiveness of the proposed model the ablation study is conducted to evaluate the importance of different designs in the architecture. We remove or adjust some modules in the network, and report their results on KITTI, as shown in Table 5.

The benefit of LGFI blocks. When all the LGFI blocks in stage 2, 3, and 4 are removed, the model size decreases by 0.4M, but the accuracy also drops. The proposed LGFI is crucial to make Mono-Lite encode long-range global contexts, thus making up for the drawback that CNNs can only extract local features.

The benefit of dilated convolutions. If all the dilation rates of convolutions in CDC blocks are set to 1, *i.e.,* there are no dilated convolutions used in the network. It can be observed that although the model size remains the same the accuracy drops more than for not using LGFI blocks. The benefit of introducing the CDC module is to enhance the locality by gradually extracting multi-scale features, while not adding additional trainable parameters.

The benefit of pooled concatenations. Accuracy also decreases when three pooled concatenations are removed.

	Encoder		Decoder		Full Model		Speed (ms)	
Method	Params. (M)	FLOPs (G)	Params. (M)	FLOPs (G)	Params. (M)	FLOPs (G)	Titan XP	Jetson Xavier
Monodepth2 [15]	11.2	4.5	3.1	3.5	14.3	8.0	3.8	14.3
R-MSFM3 [46]	0.7	2.4	2.8	14.1	3.5	16.5	7.8	22.3
R-MSFM6 [46]	0.7	2.4	3.1	28.8	3.8	31.2	13.1	41.7
MonoViT-tiny [43]	5.6	7.8	4.7	15.9	10.3	23.7	13.5	47.4
Lite-Mono-tiny (Ours)	2.0	2.4	0.2	0.5	**2.2**	**2.9**	**3.3**	**12.7**
Lite-Mono-small (Outs)	2.3	4.1	0.2	0.7	2.5	4.8	4.3	19.2
Lite-Mono (Ours)	2.9	4.4	0.2	0.7	3.1	5.1	4.5	20.0
Lite-Mono-8M (Ours)	8.1	9.5	0.6	1.7	8.7	11.2	6.5	32.2

Table 4. **Model complexity and speed evaluation.** We compare parameters, FLOPs (floating point of operations), and inference speed. The input size is 640×192, and the batch size is 16.

Architecture	Params.	Speed(ms)	Abs Rel	Sq Rel	RMSE	RMSE log	$\delta < 1.25$	$\delta < 1.25^2$	$\delta < 1.25^3$
Lite-Mono full model	3.069M	4.5	**0.107**	**0.765**	**4.561**	**0.183**	**0.886**	**0.963**	**0.983**
w/o LGFI blocks	2.661M	3.2	0.111	0.854	4.705	0.187	0.881	0.960	0.982
w/o dilated convolutions	3.069M	4.5	0.112	0.836	4.685	0.187	0.880	0.960	0.982
w/o pooled concatenations	3.062M	4.4	0.109	0.842	4.700	0.186	0.883	0.960	0.982
w/o cross-stage connections	2.942M	4.4	0.108	0.834	4.683	0.185	0.884	0.962	0.982

Table 5. **Ablation study on model architectures.** All the models are trained and tested on KITTI with the input size 640×192.

This is because when using a down-sampling layer to reduce the size of feature maps, some spatial information is also lost. The advantage of using pooled concatenations is that the spatial information is kept, and this design only adds a small number of parameters (0.007M).

The benefit of cross-stage connections. When two cross-stage connections are removed the accuracy decreases slightly. The benefit of the proposed cross-stage connections in Mono-Lite is to promote feature propagation and cross-stage information fusion.

4.7. Ablation study on dilation rates

The influence of the dilation rate in the proposed CDC module on the accuracy is studied. Four different settings are used in this experiment. **(1)** The default setting of our models is to group every three CDC blocks together, and set the dilation rate to 1, 2, and 3, respectively. For the last three blocks we set them to 2, 4, and 6. **(2)** Based on the default setting the dilation rates of the last three blocks are set to 1, 2, and 3, respectively. **(3)** Similar to the default setting we set 1, 2, and 5 in every three CDC blocks as a group. **(4)** Based on the default setting the dilation rates in the last two CDC blocks are set to 2, 4, 6 and 4, 8, 12, respectively. Table 6 lists the accuracy under different dilation settings. Comparing **(2)** with **(3)** the accuracy benefits from larger dilation rates. However, **(4)** using very large dilation rates in the late CDC blocks does not help. Simply pursuing larger dilation rates will result in the loss of local information, which is not good for the network to perceive small and medium-sized objects. Therefore, the proposed Lite-Mono adopts the setting **(1)** to extract multi-scale local features, *i.e.,* smaller dilation rates are used in shallow layers, and they are doubled in the last three CDC modules.

NO.	Abs Rel	Sq Rel	RMSE	RMSE log	δ_1	δ_2	δ_3
1	**0.107**	**0.765**	**4.561**	**0.183**	**0.886**	**0.963**	**0.983**
2	0.110	0.867	4.681	0.187	0.885	0.961	0.981
3	0.108	0.835	4.652	0.186	0.885	0.962	0.982
4	0.110	0.855	4.642	0.187	0.885	0.961	0.981

Table 6. **Ablation study on dilation rates.**

5. Conclusions

This paper presents a novel architecture Lite-Mono for lightweight self-supervised monocular depth estimation. A hybrid CNN and Transformer architecture is designed to model both multi-scale enhanced local features and long-range global contexts. The experimental results on the KITTI dataset demonstrate the superiority of our method. By setting optimized dilation rates in the proposed CDC blocks and inserting the LGFI modules to obtain the local-global feature correlations, Lite-Mono can perceive different scales of objects, even challenging moving objects closed to the camera. The generalization ability of the model is also validated on the Make3D dataset. Besides, Lite-Mono achieves a good trade-off between model complexity and inference speed.

References

[1] 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. 2, 3, 5

[2] Jimmy Lei Ba, Jamie Ryan Kiros, and Geoffrey E Hinton. Layer normalization. *arXiv preprint arXiv:1607.06450*, 2016. 5

[3] Jinwoo Bae, Sungho Moon, and Sunghoon Im. Monoformer: Towards generalization of self-supervised monocular depth estimation with transformers. *arXiv preprint arXiv:2205.11083*, 2022. 1, 2, 4, 5, 6, 7

[4] Nicolas Carion, Francisco Massa, Gabriel Synnaeve, Nicolas Usunier, Alexander Kirillov, and Sergey Zagoruyko. End-to-end object detection with transformers. In *ECCV*. Springer, 2020. 2

[5] Vincent Casser, Soeren Pirk, Reza Mahjourian, and Anelia Angelova. Unsupervised monocular depth and ego-motion learning with structure and semantics. In *CVPRW*, 2019. 1, 2

[6] Liang-Chieh Chen, George Papandreou, Florian Schroff, and Hartwig Adam. Rethinking atrous convolution for semantic image segmentation. *arXiv preprint arXiv:1706.05587*, 2017. 4

[7] 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. 6, 7

[8] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. An image is worth 16x16 words: Transformers for image recognition at scale. In *ICLR*, 2021. 1, 2, 3

[9] David Eigen and Rob Fergus. Predicting depth, surface normals and semantic labels with a common multi-scale convolutional architecture. In *ICCV*, 2015. 5, 6

[10] David Eigen, Christian Puhrsch, and Rob Fergus. Depth map prediction from a single image using a multi-scale deep network. *NeurIPS*, 27, 2014. 1, 2, 7

[11] Huan Fu, Mingming Gong, Chaohui Wang, Kayhan Batmanghelich, and Dacheng Tao. Deep ordinal regression network for monocular depth estimation. In *CVPR*, 2018. 1, 2

[12] Ravi Garg, Vijay Kumar Bg, Gustavo Carneiro, and Ian Reid. Unsupervised cnn for single view depth estimation: Geometry to the rescue. In *ECCV*, 2016. 2

[13] Andreas Geiger, Philip Lenz, Christoph Stiller, and Raquel Urtasun. Vision meets robotics: The kitti dataset. *Int. J. Robot. Res.*, 32(11):1231–1237, 2013. 2, 5, 7

[14] Clément Godard, Oisin Mac Aodha, and Gabriel J Brostow. Unsupervised monocular depth estimation with left-right consistency. In *CVPR*, 2017. 1, 2

[15] Clément Godard, Oisin Mac Aodha, Michael Firman, and Gabriel J Brostow. Digging into self-supervised monocular depth estimation. In *ICCV*, 2019. 1, 2, 3, 5, 6, 7, 8

[16] Jianyuan Guo, Kai Han, Han Wu, Yehui Tang, Xinghao Chen, Yunhe Wang, and Chang Xu. Cmt: Convolutional neural networks meet vision transformers. In *CVPR*, 2022. 2

[17] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *CVPR*, 2016. 1, 2, 3, 4, 7

[18] Dan Hendrycks and Kevin Gimpel. Bridging nonlinearities and stochastic regularizers with gaussian error linear units. *CoRR*, 2016. 4

[19] Adrian Johnston and Gustavo Carneiro. Self-supervised monocular trained depth estimation using self-attention and discrete disparity volume. In *CVPR*, 2020. 6

[20] Hyunyoung Jung, Eunhyeok Park, and Sungjoo Yoo. Fine-grained semantics-aware representation enhancement for self-supervised monocular depth estimation. In *ICCV*, 2021. 2

[21] Marvin Klingner, Jan-Aike Termöhlen, Jonas Mikolajczyk, and Tim Fingscheidt. Self-Supervised Monocular Depth Estimation: Solving the Dynamic Object Problem by Semantic Guidance. In *ECCV*, 2020. 1, 6

[22] Iro Laina, Christian Rupprecht, Vasileios Belagiannis, Federico Tombari, and Nassir Navab. Deeper depth prediction with fully convolutional residual networks. In *3DV*, 2016. 1, 2

[23] Youngwan Lee, Jonghee Kim, Jeffrey Willette, and Sung Ju Hwang. Mpvit: Multi-path vision transformer for dense prediction. In *CVPR*, 2022. 3

[24] Bo Li, Chunhua Shen, Yuchao Dai, Anton Van Den Hengel, and Mingyi He. Depth and surface normal estimation from monocular images using regression on deep features and hierarchical crfs. In *CVPR*, 2015. 2

[25] Fayao Liu, Chunhua Shen, Guosheng Lin, and Ian Reid. Learning depth from single monocular images using deep convolutional neural fields. *IEEE TPAMI*, 38(10):2024–2039, 2015. 2

[26] Ilya Loshchilov and Frank Hutter. SGDR: Stochastic gradient descent with warm restarts. In *ICLR*, 2017. 6

[27] Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. In *ICLR*, 2018. 6

[28] Xiaoyang Lyu, Liang Liu, Mengmeng Wang, Xin Kong, Lina Liu, Yong Liu, Xinxin Chen, and Yi Yuan. Hr-depth: High resolution self-supervised monocular depth estimation. In *AAAI*, 2021. 1, 6, 7

[29] Muhammad Maaz, Abdelrahman Shaker, Hisham Cholakkal, Salman Khan, Syed Waqas Zamir, Rao Muhammad Anwer, and Fahad Shahbaz Khan. Edgenext: Efficiently amalgamated cnn-transformer architecture for mobile vision applications. In *CADL*, 2022. 2

[30] Matteo Poggi, Filippo Aleotti, Fabio Tosi, and Stefano Mattoccia. On the uncertainty of self-supervised monocular depth estimation. In *CVPR*, 2020. 2

[31] René Ranftl, Alexey Bochkovskiy, and Vladlen Koltun. Vision transformers for dense prediction. In *ICCV*, 2021. 2

[32] Ashutosh Saxena, Min Sun, and Andrew Y Ng. Make3d: Learning 3d scene structure from a single still image. *IEEE TPAMI*, 31(5):824–840, 2008. 2, 5, 7

[33] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014. 1, 2

[34] Robin Strudel, Ricardo Garcia, Ivan Laptev, and Cordelia Schmid. Segmenter: Transformer for semantic segmentation. In *ICCV*, 2021. 2

[35] Arnav Varma, Hemang Chawla, Bahram Zonooz, and Elahe Arani. Transformers in self-supervised monocular depth estimation with unknown camera intrinsics. *arXiv preprint arXiv:2202.03131*, 2022. 1, 2, 3

[36] Chaoyang Wang, José Miguel Buenaposada, Rui Zhu, and Simon Lucey. Learning depth from monocular videos using direct methods. In *CVPR*, 2018. 6, 7

[37] Zhou Wang, Alan C Bovik, Hamid R Sheikh, and Eero P Simoncelli. Image quality assessment: from error visibility to structural similarity. *IEEE TIP*, 13(4):600–612, 2004. 5

[38] Dan Xu, Elisa Ricci, Wanli Ouyang, Xiaogang Wang, and Nicu Sebe. Multi-scale continuous crfs as sequential deep networks for monocular depth estimation. In *CVPR*, 2017. 2

[39] Jiaxing Yan, Hong Zhao, Penghui Bu, and YuSheng Jin. Channel-wise attention-based network for self-supervised monocular depth estimation. In *3DV*, 2021. 1, 2, 6

[40] Nan Yang, Lukas von Stumberg, Rui Wang, and Daniel Cremers. D3vo: Deep depth, deep pose and deep uncertainty for monocular visual odometry. In *CVPR*, 2020. 2

[41] Zhichao Yin and Jianping Shi. Geonet: Unsupervised learning of dense depth, optical flow and camera pose. In *CVPR*, 2018. 1, 2, 6

[42] Fisher Yu and Vladlen Koltun. Multi-scale context aggregation by dilated convolutions. *arXiv preprint arXiv:1511.07122*, 2015. 3

[43] Chaoqiang Zhao, Youmin Zhang, Matteo Poggi, Fabio Tosi, Xianda Guo, Zheng Zhu, Guan Huang, Yang Tang, and Stefano Mattoccia. Monovit: Self-supervised monocular depth estimation with a vision transformer. In *3DV*, 2022. 3, 6, 7, 8

[44] Hang Zhou, David Greenwood, and Sarah Taylor. Self-supervised monocular depth estimation with internal feature fusion. *arXiv preprint arXiv:2110.09482*, 2021. 1, 2, 3, 5

[45] Tinghui Zhou, Matthew Brown, Noah Snavely, and David G Lowe. Unsupervised learning of depth and ego-motion from video. In *CVPR*, 2017. 1, 2, 5

[46] Zhongkai Zhou, Xinnan Fan, Pengfei Shi, and Yuanxue Xin. R-msfm: Recurrent multi-scale feature modulation for monocular depth estimating. In *ICCV*, 2021. 1, 2, 3, 5, 6, 7, 8