Pattern Analysis and Applications (2023) 26:1833–1846
https://doi.org/10.1007/s10044-023-01205-4

SHORT PAPER

EMTNet: efficient mobile transformer network for real-time monocular depth estimation

Long Yan[1] · Fuyang Yu[2] · Chao Dong[2]

Received: 16 April 2023 / Accepted: 6 September 2023 / Published online: 7 October 2023

Abstract
Estimating depth from a single image presents a formidable challenge due to the inherently ill-posed and ambiguous nature of deriving depth information from a 3D scene. Prior approaches to monocular depth estimation have mainly relied on Convolutional Neural Networks (CNNs) or Vision Transformers (ViTs) as the primary feature extraction methods. However, striking a balance between speed and accuracy for real-time tasks has proven to be a formidable hurdle with these methods. In this study, we proposed a new model called EMTNet, which extracts feature information from images at both local and global scales by combining CNN and ViT. To reduce the number of parameters, EMTNet introduces the mobile transformer block (MTB), which reuses parameters from self-attention. High-resolution depth maps are generated by fusing multi-scale features in the decoder. Through comprehensive validation on the NYU Depth V2 and KITTI datasets, the results demonstrate that EMTNet outperforms previous real-time monocular depth estimation models based on CNNs and hybrid architecture. In addition, we have done the corresponding generalizability tests and ablation experiments to verify our conjectures. The depth map output from EMTNet exhibits intricate details and attains a real-time frame rate of 32 FPS, achieving a harmonious balance between real-time and accuracy.

Keywords Deep learning · Vision transformer · Monocular depth estimation · Real-time task · Attention mechanism

1 Introduction

Estimating depth from a single image is a task that humans can accomplish easily, but achieving high precision and low resource requirements with computational models is notoriously difficult. Depth estimation is a fundamental problem in computer vision that is significant for various applications such as scene understanding [1], robot navigation, autonomous driving, augmented reality, scene 3D reconstruction [2], and obstacle detection [3].

Monocular depth estimation (MDE) is the task of obtaining depth information for each pixel from a single RGB image. It is a challenging task because obtaining 2D depth information from a 3D scene is an inherently ill-posed and ambiguous problem. A single 2D depth image can be generated from an infinite number of 3D scenes [4]. Furthermore, retrieving depth information without the assistance of additional data, such as stereo images, optical flow, point clouds, and other data, is extremely difficult.

While devices such as depth cameras and LIDAR can directly obtain depth information, they can be quite expensive. An alternative approach is to use binocular images and video sequences to estimate depth [5–8]. However, stereo matching based on binocular vision requires pixel-by-pixel correspondence and disparity calculation, resulting in higher computational complexity for matching. Moreover, a single pixel may match numerous identical feature points in low-texture scenes, leading to poor matching outcomes. In contrast, monocular depth estimation is relatively less expensive and more easily accessible. With the development of convolutional neural networks (CNNs), monocular depth estimation methods based on CNNs have emerged as

✉ Long Yan
yanlong@sdtbu.edu.cn

Fuyang Yu
fuyang.yu@sdtbu.edu.cn

Chao Dong
2021420073@sdtbu.edu.cn

[1] School of Management Science and Engineering, Shandong Technology and Business University, Yantai, China

[2] School of Information and Electronic Engineering, Shandong Technology and Business University, Yantai, China

an alternative to earlier methods that relied on manually created features [9–11].

Previous depth estimation methods have heavily relied on CNN-based techniques, which have significantly improved accuracy. However, CNN-based methods are not always able to make accurate estimates for complex scenes or areas with missing depth information. To address these challenges, researchers have attempted to increase the depth of the model to expand the receptive field of convolution and improve feature extraction capabilities. However, increasing the depth of the model also leads to an increase in the number of parameters, making the model larger and more resource-intensive.

The recent proposals of Transformer [12] and ViT [13] have led to a new approach in computer vision. ViT uses self-attention to learn global information in images for various vision tasks, and there are many ViT-based models in monocular depth estimation [14–19]. ViT is well-suited for extracting global features in vision tasks, but it also makes the model larger, slower to infer an image, and more difficult to train.

In this paper, we propose the Efficient Mobile Transformer Network (EMTNet) for real-time scene depth estimation show in Fig. 1. Inspired by MoCoViT [20], we use the mobile transformer block (MTB) to reuse redundant parameters in self-attention calculations, reducing the number of parameters and improving the real-time performance of the model. The EMTNet encoder utilizes both CNN and ViT architectures to extract deep features from local and global scales. Furthermore, we use the DPT [21] decoder to restore resolution and fuse multi-scale depth information to produce high-resolution depth maps.

To validate the performance of our proposed model, we evaluated it on two monocular depth estimation datasets, NYU Depth V2 (indoor dataset, depth range 0-10 m) and KITTI (outdoor dataset, depth range 0-80 m), with corresponding training configurations. Our experiments demonstrate that our depth map output has higher resolution and finer detail than other techniques. Moreover, our method achieves a frame rate of 32 FPS in real-time while maintaining high accuracy depth map output.

The main contributions of this paper are as follows:

- Proposed a new model for real-time monocular depth estimation based on MTB and named EMTNet. MTB uses the Branch Sharing scheme to simplify the computation of the attention graph, thus reducing the number of parameters in the model and achieving real-time detection. It achieves a harmonious balance between real-time capability and minimal parameters within the same architecture.
- In order to enhance the feature extraction capability, we construct the encoder segment of the model by combining the CNN and ViT architectures. The encoder acquires deep features from two scales, local and global, respectively, which greatly improves the model's ability to capture deep information.

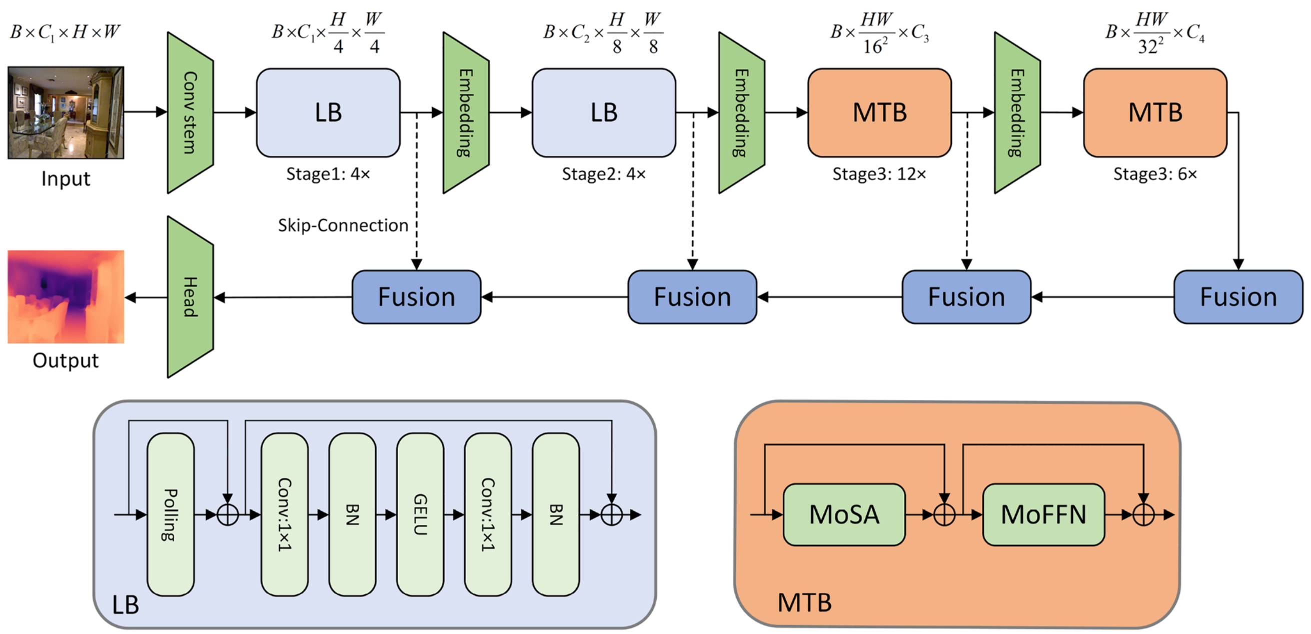
$B \times C_1 \times H \times W$
Input
Conv stem
$B \times C_1 \times \frac{H}{4} \times \frac{W}{4}$
LB
Stage1: 4×
Embedding
$B \times C_2 \times \frac{H}{8} \times \frac{W}{8}$
LB
Stage2: 4×
Embedding
$B \times \frac{HW}{16^2} \times C_3$
MTB
Stage3: 12×
Embedding
$B \times \frac{HW}{32^2} \times C_4$
MTB
Stage3: 6×
Skip-Connection
Fusion
Fusion
Fusion
Fusion
Head
Output
Polling
Conv:1×1
BN
GELU
Conv:1×1
BN
LB
MoSA
MoFFN
MTB

Fig. 1 The overall framework of EMTNet. Our architecture consists of two major components: the encoder section for extracting depth features and the decoder section for fusing features at each scale, where the encoder consists mainly of Linear Block (LB) and Mobile Transformer Block (MTB). The diagrams of LB and MTB are shown below the overall architecture diagram, from left to right

- We conducted experiments on two public datasets, NYU Depth V2[22] and KITTI[23]. The experimental results showed that our method achieved better results in all equivalent architectural models. Meanwhile, our model outperformed other methods in the output prediction results, and it also contained more obvious detail information in complex scenes.

The remainder of this work has been structured in the following manner. Section 2 consists of a literature review pertaining to this research direction, where the research conducted in the field of depth estimation over the past few years is discussed. Section 3 has provided a detailed description of the architecture of the network that was proposed in this paper, along with its implementation specifics. In order to validate the efficacy of our approach, several experiments have been carried out in Sect. 4, and their outcomes have been discussed. Section 5 addresses the limitations of our methodology and outlines potential avenues for future research. Ultimately, the results of this work have been summarized in Sect. 6.

2 Related work

In this section, we will introduce the research background in the field of monocular depth estimation and Vision Transformer, and summarize the methods used in some of the past previous work.

2.1 Monocular depth estimation

Depth estimation from a single color image has been an active area of research in robotics and computer vision for more than a decade. Early methods relied on hand-crafted features and probabilistic graphical models to estimate depth from RGB images captured by monocular cameras. For example, Saxena et al. [24] estimated the absolute scales of different image patches and inferred depth using a Markov random field model. Nonparametric methods [25–28] have also been used to estimate depth by combining the depth of the image with similar photometric content retrieved from a database. In recent years, depth estimation has shifted toward modern deep learning-based methods [29–31], replacing manual feature representations with learned features extracted from neural networks.

The state-of-the-art methods for depth estimation from RGB images involve training convolutional neural networks using large-scale datasets. For example, Eigen et al. were among the first to use deep learning for this task [29]. They proposed a two-stack CNN approach, where one stack predicts the global coarse scales and the other stack refines local details, using fine-scale networks for more accurate depth maps. Eigen and Fergus [9] further incorporated auxiliary prediction tasks into the architecture. Liu et al. [32] combined a deep CNN with a continuous conditional random field to obtain sharper transitions and local details. Laina et al. [30] developed a deep residual network based on ResNet [33] and achieved even higher accuracy than previous methods. To solve the ambiguity problem in prediction, Qi et al. [34] trained their network to estimate both depth and normals.

Depth estimation is commonly addressed as a dense prediction regression problem, but recent research has explored treating it as a classification problem. This involves dividing the depth range into multiple bins and predicting which bin each pixel belongs to. Fu et al. [31] pioneered this approach by utilizing ordinal regression to convert the depth estimation problem into a classification problem. Bhat et al. [14] uses adaptive bins and a lightweight neural network to estimate depth probability distributions, which are then combined to generate the final depth map. Li et al. [15] built on this approach and incorporated full interaction between the probability distribution and bins, using Transformer to generate bins. While predicting depth in discrete bins can simplify training with limited data, it may reduce accuracy compared to predicting continuous values, and the number of bins used can also impact accuracy.

Another promising approach to depth estimation is to use a ViT-based architecture. The ViT [13] is a deep learning model that allows the utilization of global features for a wide range of computer vision tasks. In recent years, many researchers have proposed ViT-based methods for monocular depth estimation. For example, Bhat et al. [14] and Li et al. [15] both incorporated ViT into their method to improve the accuracy of depth estimation. Other studies, such as Zhao et al. [16], Bae et al. [17], Li et al. [18], and Shu et al. [19], have also proposed ViT-based methods that achieve state-of-the-art performance on monocular depth estimation benchmarks. These methods generally leverage the attention mechanism of the ViT to capture global context information and combine it with local features to improve the accuracy of depth estimation.

2.2 Vision transformer

The Vision Transformer is a neural network architecture that has shown promising results in computer vision, leading to the emergence of many works based on ViT. For example, DeiT [35] uses knowledge distillation based on ViT [13] to train a small model that achieves accuracy comparable to that of a larger model. PVT [36] uses a pyramid attention mechanism to handle features at different scales, and a cross-layer feature pyramid to improve feature representation. TNT [37] uses a spatial Transformer network and dynamic convolution to improve the deformability and receptive field of the model. CoaT [38] is a multi-layered network structure that uses multi-scale features and multi-layered attention mechanisms to handle features at different levels. Finally, Swin Transformer [39] uses an interleaved local attention mechanism and a global attention mechanism to handle images with relatively large aspects.

Moreover, a number of lightweight ViT models have been proposed to address real-time applications. For instance, ResViT [40] suggests an improved residual connection method to further reduce the computational burden of the lightweight ViT model. MobileViT [41] introduces a lightweight ViT model for mobile devices, which delivers faster inference speed and smaller model size on mobile devices. LViT [42] achieves good performance by reducing the number of model parameters and computational complexity through the removal of unnecessary modules and downsampling of resolution. Lastly, TinyViT [43] employs grouped convolution and depth-separable convolution to reduce the number of parameters and computational complexity of the model, thus enabling efficient image classification by introducing the transformer module into the conventional neural network.

Recent research [44] has demonstrated that combining convolution and Transformer can enhance prediction accuracy and improve training stability. BoTNet [45] achieved significant advancements in instance segmentation and object detection by replacing the last three bottleneck blocks of ResNet [33] with self-attention. ConViT [46] improved ViT with soft convolutional induction bias by introducing gated position self-attention (GPSA). The CVT [47] combines CNNs with ViT to improve computer vision tasks by introducing localized convolutional operations. LeViT [48] proposes a lightweight ViT model based on the LeNet [49] architecture. In this paper, we adopt the approach of combining CNN networks and Transformer by incorporating the MTB self-attention module into a CNN network, which enhances the model's feature extraction capabilities and real-time depth estimation performance.

3 Architecture

In this section, we will introduce the overall structure of EMTNet and explain its principles accordingly, which includes encoder and decoder parts. After that, we will introduce the implementation details of the Mobile Transformer Block (MTB), which includes Mobile Self-Attention (MoSA) and Mobile Feed Forward Network (MoFFN) specifically designed for lightweight networks. Finally, we introduce the loss function we used.

3.1 Overview of the network

Although previous CNN-based methods excel at extracting deep feature information from images, they are limited in capturing only localized features due to their restricted receptive fields. Achieving an expanded receptive field typically involves stacking multi-layer CNNs or using dilated convolutions, which inevitably leads to an increase in model parameters or loss of feature information. Our proposed model, on the other hand, capitalizes on the strengths of both CNNs and ViTs. By combining these two architecture, our model effectively extracts depth feature information at both local and global scales, substantially enhancing the overall feature extraction capability. To restore depth information for monocular depth estimation tasks, we employ a fusion module with a skip connection. This module fuses depth information from multiple scales, helping to preserve feature information at each scale during picture restoration. The overall architecture of our proposed network is illustrated in Fig. 1.

Our network follows a standard Encoder-Decoder architecture, comprising an encoder for extracting depth features and a decoder for fusing multi-scale features. The encoder comprises four stages, the first two of which are Linear Block (LB) that extract local features in the scene using a Conv-net style. The latter two stages use Mobile Transformer Block (MTB), which includes two sub-modules, MoSA and MoFFN, designed to extract global information from the scene while reducing computational effort.

EMTNet Encoder. This part is used to extract depth features. It contains a total of four stages, each consisting of N_i identical modules, and we set N_1, N_2, N_3 and N_4 to 4, 4, 12 and 6 respectively. First, input images are processed by a CONV stem with two 3×3 convolutions with stride 2 as patch embedding, which is used to speed up the subsequent model processing,

$$X_1^{B,C_{j|j=1},\frac{H}{4},\frac{W}{4}} = \mathrm{PatchEmbed}\left(X_0^{B,3,H,W}\right), \tag{1}$$

where C_j represents the number of channels in the jth stage, X_0 and X_1 denote the input of the image and the output of the CONV stem, respectively. The X_1 then fed into the first stage, where we use four LB to initially extract the feature information. the structure of the LB is shown in Fig. 1, it starting with a pooling layer to extract the low-level features,

$$\begin{aligned} I_i &= \mathrm{Pool}\left(X_i^{B,C_j,\frac{H}{2^{j+1}},\frac{W}{2^{j+1}}}\right) + X_i^{B,C_j,\frac{H}{2^{j+1}},\frac{W}{2^{j+1}}}, \\ X_{i+1}^{B,C_j,\frac{H}{2^{j+1}},\frac{W}{2^{j+1}}} &= \mathrm{Conv}_B\left(\mathrm{Conv}_{B,G}\left(I_i\right)\right) + I_i, \end{aligned} \tag{2}$$

where Pool indicates a pooling operation, $\mathrm{Conv}_{\mathrm{B,G}}$ denotes a subsequent convolution containing both BN and GELU operations, and $\mathrm{Conv}_{\mathrm{B}}$ denotes a subsequent convolution containing only BN operations.

The second stage of the operation is similar to the first stage, with the use of 4 LB blocks. However, after passing through the Embedding layer, the feature map becomes half the size but with twice the number of channels.

In the third and fourth stages, we employ 12 and 6 MTBs respectively to extract global information from the features. The MTBs are constructed by modifying traditional self-attention, and consist of two sub-modules: MoSA and MoFFN.

$$\begin{aligned} I_i &= \mathrm{MoSA}\left(X_i^{B,\frac{HW}{4^{j+1}},C_j}\right) + X_i^{B,\frac{HW}{4^{j+1}},C_j}, \\ X_{i+1}^{B,\frac{HW}{4^{j+1}},C_j} &= \mathrm{MoFFN}\left(I_i\right) + I_i, \end{aligned} \tag{3}$$

where i denotes the tokens entered by the ith MTB and j denotes the module operation at stage j.

We decided to place LB before MTB in our network architecture based on the intuition that LB is better suited for extracting local feature information for constructing edge contours, while MTB is better suited for extracting global features to estimate continuous large areas. In monocular depth estimation, the contour information of objects is particularly important as the most distinct depth variation is often found at the edges of the object, while the depth variation is smoother or more consistent inside the object contour. Although CNN-based models do expand the receptive field of convolution when dealing with higher dimensional features and to some extent use global information, they do not perform as well as MTB in extracting global information. Therefore, placing LB before MTB was based on our consideration of feature extraction. In the following sections, we will describe the overall structure of the EMTNet in more detail.

EMTNet Decoder. We designed the Fusion module for fusing intermediate features from the previous Fusion module and the corresponding encoder stage. The use of skip connections in the Fusion modules allows for the preservation of feature information at multiple scales, preventing information loss during image restoration. The final output feature map is 1/4 the size of the original map after being downsampled by the four Fusion modules. To produce the final depth prediction, we added an output header dedicated to depth estimation. The detailed structure of the decoder is shown in Fig. 2.

The Fusion module starts with a convolutional layer to adjust the dimensionality of the feature map, which we set to remain the same before and after the convolution in our implementation. We also plan to incorporate depthwise separable convolution (DSC) in subsequent ablation experiments to test our hypothesis. The output of the convolutional layer then undergoes a Residual Conv Unit, which is added to the output of the previous module. The result is passed through another Residual Conv Unit, followed by upsampling and linear projection for the final output to the next module. To ensure better performance, we use the GELU activation function instead of the ReLU activation function in the Fusion module, as verified in the ablation experiment Sect. 4.4. In the following section, we will delve into the specifics of the MTB.

3.2 Mobile transformer block

Although CNN-based methods can extract depth features, they are limited in their ability to extract global feature information due to their inherent characteristics. The Transformer [12] and ViT [13] were proposed to address this limitation by enabling models to extract features by combining global information from the image, thus improving generalization ability and accuracy. However, these approaches using

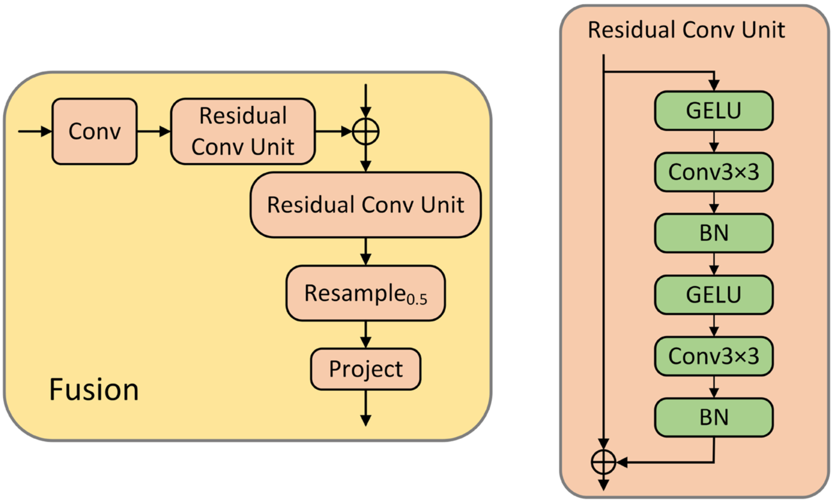

Fig. 2 The overview of Fusion module. The Fusion module consists mainly of a CNN architecture for fusing features at different scales and upsampling the output to the next module

global attention have a significantly larger number of parameters compared to CNN-based models, making them less suitable for real-time applications. In our proposed method, we enhance traditional self-attention by introducing MTB to reduce the number of parameters and FLOPs of the model, enabling real-time monocular depth estimation with improved accuracy. Self-attention used in the traditional ViT is,

$$\text{Self-Attention}(Q, K, V) = \text{Softmax}\left(\frac{QK^T}{\sqrt{d_k}}\right)V, \tag{4}$$

where Q, K and V are the three matrices that can be learned by model. The Q and K matrices operations take up the majority of the model's processes for the calculation of self-attention, so this is where the MTB module needs to be improved the most.

While self-attention is a powerful mechanism for capturing global dependencies in an image, it becomes less advantageous than convolutional layers in lightweight models with constrained capacity due to its quadratic computational complexity with relation to spatial resolution. To compute a linear combination of results for V, traditional self-attention requires three linear layers of the same level. When dealing with multi-head self-attention, the superposition of multiple self-attentions significantly increases the number of parameters. To address this problem, we introduce MoSA, which replaces traditional self-attention with an attention mechanism specifically designed for lightweight Transformer structures.

Mobile Self-Attention (MoSA). MoSA uses a branch sharing scheme to reuse weights in the Q, K and V calculations, making it an attention mechanism designed for lightweight Transformer structures. As shown in Fig. 3, F^q, F^k and F^v are projections of Q, K and V with the same input features, respectively. The approach reuses the features V directly into Q and K based on the intuition that Q and K are only involved in the computation of the attention graph, while the final result of the self-attention mechanism is a linear combination of each token in V. Thus, V must retain more semantic information than Q and K to guarantee the final weight and representational power of the results. As a result, the correlation between the results of self-attention and V is stronger than their correlation with Q and K. This simplifies the computation of Q and K for real-time tasks, achieving a better balance of performance overhead. Compared to traditional self-attention, MoSA replaces the Q and K matrices with the V matrix, resulting in fewer parameters and faster computation.

$$\begin{aligned} F^v &= F^k = F^q \\ V &= F^v(X) \\ K &= F^k(X) = V^T \\ Q &= F^q(X) = V \end{aligned} \tag{5}$$

where F^v, F^q and F^k are projections used to compute V, Q and K, respectively. To avoid feature loss, a depth-separable

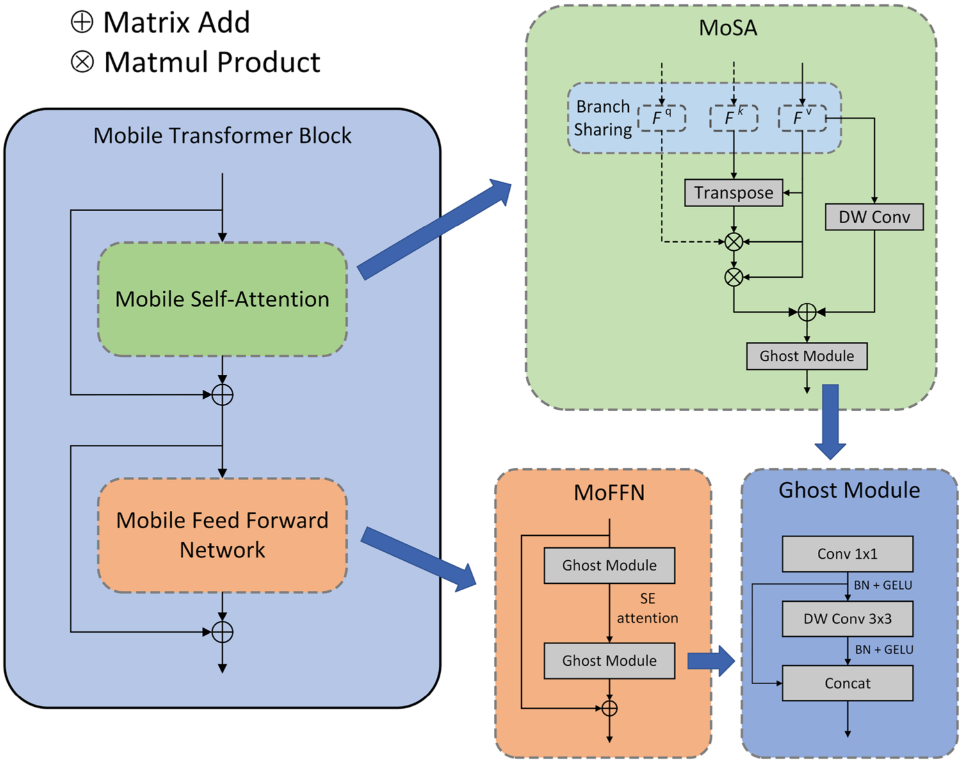

Fig. 3 The Mobile Transformer Block. Mobile Transformer Block consists of Mobile Self-Attention (MoSA) and Mobile Feed Forward Network (MoFFN). The branch sharing mechanism in MoSA avoids computing Q and K, and computes the attention map by reusing V. Ghost module is used to replace Linear layer, and LayerNorm is removed for efficiency

convolution branch is added to the output of V. The improved self-attention is,

$$\mathrm{MoSA}(V) = \mathrm{Softmax}\left(\frac{VV^T}{\sqrt{d_k}}\right)V + \mathrm{Depthwise}(V) \tag{6}$$

Mobile Feed Forward Network (MoFFN). MoFFN is a fine-grained feature operation that replaces the linear layer in traditional self-attention with a more efficient Ghost [50] module. To extract features on the channels, MoFFN contains two Ghost modules with a Squeeze-and-Excitation Networks (SENet) [51] inserted between them. In image processing tasks, channel domain attention explicitly models the interdependencies between feature channels. Following the suggestion of Hu et al. [51], we placed the SENet inside the model of the residual structure, and the module ends with a residual connection.

Figure 3 illustrates the MoFFN module and the Ghost structure, a widely used technique in lightweight networks for constructing features in a cost-effective manner. The Ghost module uses standard convolution to generate a few intrinsic feature maps, which are then expanded to a larger number of channels using cheap linear operations. To achieve a better balance between performance and speed, these linear operations are typically implemented as depthwise convolutions. MoFFN can be expressed as follows,

$$\begin{aligned} y &= \mathrm{Ghost}(\mathrm{SE}(\mathrm{Ghost}(x))) + x \\ \mathrm{Ghost}(x) &= \mathrm{Concat}\left[\mathrm{DWConv}_{B,G}\left(\mathrm{Conv}_{B,G}(x)\right), \mathrm{Conv}_{B,G}(x)\right] \end{aligned} \tag{7}$$

where SE denotes the channel attention module, $\mathrm{DWConv}_{B,G}$ denotes the subsequent depthwise separable convolution containing BN and GELU operations, and $\mathrm{Conv}_{B,G}$ denotes the subsequent ordinary convolution containing BN and GELU operations.

The Ghost module is a widely acknowledged structure in lightweight networks, and its effectiveness has been extensively demonstrated. The MoFFN, consisting of the Ghost module and SENet [51], serves as an efficient replacement for the traditional Feed-Forward Network (FFN) in ViT. The MoFFN module proves highly effective in addressing real-time tasks.

MoSA and MoFFN together constitute the MTB. In this work, we leverage MTB to reduce the computational load of the model, enhancing its efficiency while preserving model accuracy. The primary objective of adopting MTB is to streamline attention computation and improve the real-time speed of the model. Significantly, the computational speed of MTB outperforms that of the traditional self-attention module. In the forthcoming experimental section, we will comprehensively compare the information processing speed of models featuring different architectures, providing a comprehensive analysis of their respective performances.

Loss function: Inspired by [52], we use the Scale-Invariant (SI) loss proposed by Eigen et al. [29] as our loss function,

$$\ell_{pixel} = \alpha\sqrt{\frac{1}{T}\sum_i g_i^2 - \frac{\lambda}{T^2}\left(\sum_i g_i\right)^2} \tag{8}$$

where $g_i = \log\tilde{d}_i - \log d_i$, d_i is the ground truth depth, $\tilde{d}_i$ is the estimated depth and T denotes the number of pixels having valid ground truth values. We use $\lambda = 0.85$ and $\alpha = 10$ for all our experiments.

4 Experiments

We have conducted extensive experiments on standard depth estimation for single image datasets for both indoor and outdoor scenes. In the following, the first section begins with a brief description of the individual datasets and the evaluation metrics. The second section describes the implementation details of the experiments. In the third part, we compare the model quantitatively with previous monocular depth estimation methods and perform generalizability tests. In the fourth section, we conduct ablation experiments to validate the effectiveness of our network. In the last section, we summarize and analyze all the experimental results and discuss the final results of the experiments.

4.1 Datasets and evaluation metrics

We tested the model on three datasets and in that section the datasets and the treatment of the data are presented. The evaluation metrics used is presented at the end.

NYU Depth V2 is a dataset that provides images and depth maps of various indoor scenes captured at a pixel resolution of 640 × 480 [22]. The dataset comprises 1,449 densely labeled images and 407,024 pseudo-labeled and unlabeled images. We used a subset of 24,231/654/654 images for training, validation, and testing. During the training period, we preprocessed the original data through random cropping and rotation, using a crop size of 416×544. We used the original image size of 480×640 during the testing period.

KITTI is a dataset that provides stereo images and corresponding 3D laser scans of outdoor scenes captured using equipment mounted on a moving vehicle [23]. The KITTI dataset contains real image data collected from urban, rural, and highway scenes, with a sampling resolution of 375× 1,242, sampled and synchronized at 10Hz. We selected the depth prediction dataset as our model test data, using a subset of 23,158/697/697 images for training, validation, and testing, respectively. During the training period, we performed image enhancement on the original data through random cropping and rotation in the image preprocessing step. We used a crop size of 320×1,056 for the random cropping operation. We did not use the same preprocessing operation for validation, and we used the original size of 375×1,242 for both validation and testing.

SUN RGB-D is a publicly available dataset on scene understanding from the Vision & Robotics Group at Princeton University. SUN RGB-D is captured by four different sensors and contains 10,000 RGB-D images, at a similar scale as PASCAL VOC [53]. The whole dataset is densely annotated and includes 146,617 2D polygons and 58,657 3D bounding boxes with accurate object orientations, as well as a 3D room layout and category for scenes. This dataset enables us to train data-hungry algorithms for scene-understanding tasks, evaluate them using direct and meaningful 3D metrics, avoid overfitting to a small testing set, and study cross-sensor bias. We use this dataset as a benchmark of model generalization ability for determining the generalization results of different models trained on NYU Depth V2.

Evaluate metrics. To evaluate the accuracy of the depth estimation results, we used the generalized standard metric for depth estimation proposed by Eigen et al. [29]. These error metrics are defined as:

- threshold accuracy (δ_i): % of y_p s.t. $\max(\frac{y_p}{\hat{y}_p}, \frac{\hat{y}_p}{y_p}) = \delta < thr$ for $thr = 1.25, 1.25^2, 1.25^3$;
- absolute relative error (AbsRel): $\frac{1}{n}\sum_p^n \frac{|y_p - \hat{y}_p|}{y}$;
- squared relative error (SqRel): $\frac{1}{n}\sum_p^n \frac{\|y_p - \hat{y}_p\|^2}{y}$;
- root mean squared error (RMSE): $\sqrt{\frac{1}{n}\sum_p^n (y_p - \hat{y}_p)^2}$;
- root mean squared log error (RMSE_{log}): $\sqrt{\frac{1}{n}\sum_p^n \left\| log(y_p) - log(\hat{y}_p) \right\|^2}$;

where y_p is a pixel in depth image y, $\hat{y}_p$ is a pixel in the predicted depth image $\hat{y}$, and n is the total number of pixels for each depth image.

4.2 Implementation details

We implemented the proposed method using PyTorch version 1.12 on Ubuntu 20.04, and trained it on a single NVIDIA GeForce RTX 2080 Ti graphics card. Prior to inputting the original image into the model, we applied standard data augmentation and image enhancement techniques to the image. The specific methods are as follows:

- Cropping: both the input image and the target image were randomly cropped. For the NYU Depth V2 dataset, the image was cropped to a size of 416×544, and for the KITTI dataset, the image was cropped to a size of 320×1056.
- Rotation: we randomly rotated the input image and target image between the angles $r \in [-2.5, 2.5]$.
- Gamma enhancement: we applied gamma enhancement to the original image with γ powers. The value of γ was randomly selected from the range $\gamma \in [0.9, 1.1]$.
- Brightness enhancement: we multiplied the original image with d to create a random variation in brightness. For the NYU Depth V2 dataset, the value of d was randomly selected from the range $d \in [0.75, 1.25]$, and for the KITTI dataset, the value of d was randomly selected from the range $d \in [0.9, 1.1]$.
- Color enhancement: we multiplied the original image by a random RGB value $c \in [0.9, 1.1]$.
- Horizontal flip: we randomly flipped the image and target image horizontally with a probability of 0.5.

In addition to the above operations, several training techniques were used to accelerate the convergence of the model. The training method was set as follows.

We employed the AdamW optimization algorithm with weight decay 0.1 to update the parameters of our models during the training period. The maximum learning rate for the NYU Depth V2 and KITTI datasets was set to 5×10^{-4} and 3.75×10^{-4}, respectively. To accelerate the convergence of the model, we also employed learning rate warm-up [33] and OneCycleLR policy [54]. Specifically, we set the learning rate with $max\_lr = 3.5 \times 10^{-4}$ and warm-up the learning rate from $max\_lr/25$ to max_lr for the first 30% of iterations, followed by cosine annealing to $max\_lr/100$. The number of training epochs was set to 200, with a batch size of 10, until the model finally converged and stopped. Training our model took approximately 25 min per epoch on a single node with one NVIDIA GeForce RTX 2080 Ti graphics card. Finally, we validated each dataset after training and tested the test set with the best model on the validation set.

4.3 Comparison with the most advanced available

We assess the efficacy of our proposed network through rigorous evaluations on two datasets: the NYU Depth V2 and KITTI. To ensure a comprehensive appraisal, we judiciously curate pertinent methodologies hitherto applied in real-time monocular depth estimation. These established approaches serve as benchmarks for comparative analysis, affording invaluable insights into the performance of our architectural model. Furthermore, we conduct expansive generalizability assessments on the SUN RGB-D dataset, alongside meticulous ablation experiments aimed at validating the model's robustness.

Results on NYU Depth V2. Table 1 presents a comprehensive performance comparison on the official NYU Depth V2

Table 1 Comparison of performances on the NYU Depth V2. The reported numbers are from the corresponding original papers. Measurements are made for the depth range from 0 m to 10 m. The best results are in **bold**, second best are underlined

Methods	$\delta_1 \uparrow$	$\delta_2 \uparrow$	$\delta_3 \uparrow$	AbsRel↓	SqRel↓	RMSE↓	$\text{RMSE}_{log} \downarrow$
Nekrasov *el al.*[55]	**0.816**	0.952	**0.989**	0.142	0.127	**0.483**	**0.196**
Eigin *el al.* [29]	0.719	0.860	0.941	0.223	–	0.687	0.260
FastDepth[56]	0.731	0.926	0.978	0.175	0.149	0.605	0.216
FastDepth V2[56]	0.778	0.946	0.986	0.156	0.116	0.539	0.210
CReaM[57]	0.704	0.917	0.977	0.190	–	0.687	0.251
DepthNet [58]	0.807	0.949	0.983	0.144	0.105	0.599	–
Ma *el al.*[59]	0.681	0.899	0.969	0.201	0.187	0.667	–
Yucel *el al.*[60]	0.775	–	–	–	–	0.599	–
An *el al.*[61]	0.803	**0.953**	0.987	0.146	0.106	0.514	0.199
EMTNet (Ours)	0.815	**0.953**	**0.989**	**0.141**	**0.099**	0.490	0.199

test set. Our model outperforms previous methods on most metrics, showcasing its superiority. However, we noticed that it does not exhibit significant improvement on certain metrics, such as RMSE and RMSElog, and only marginal enhancements on others. We speculate that one of the reasons why our method did not show a strong advantage on this dataset may be that the MTB module is too much on the speed side and somewhat less on the accuracy side, which is inevitable. Nevertheless, Fig. 4 visually illustrates the depth prediction results of our model alongside its comparison model, emphasizing subtle differences in the same scene with white dashed boxes. Despite not necessarily dominating in quantity, our method excels in the quality of generated depth maps, particularly in capturing finer detail information within complex scenes. Moreover, our model exhibits remarkable depth completion ability in regions with missing depth compared to other models. In contrast, outputs from comparison models like An et al.[61] and Wofk et al.[56] exhibit erroneous estimations in depth-missing regions, rendering them nearly indistinguishable from the surrounding scene information. These findings highlight the strength of our model in producing accurate and detailed depth maps, especially in challenging scenarios.

Results on KITTI. Table 2 provides an overview of the performance metrics for all related models on the KITTI test set. Our model emerges as the clear leader, showcasing significantly superior results across all evaluated metrics. Particularly noteworthy is its outstanding performance when compared to methods like MonoFormer, Lite-Mono, and Varma et al., which also utilize the Transformer architecture. Quantitatively, our approach obviously outperforms these Transformer-based methods. Figure 5 illustrates the depth prediction results for our model alongside the comparison methods. Focusing on areas highlighted by white dashed boxes in the figure, our model excels in capturing intricate details and exhibits exceptional depth prediction accuracy, especially in regions with missing depth information.

Results on SUN RGB-D. Table 3 presents the results of the generalization tests on SUN RGB-D. For the assessment of generalization ability, we carefully selected a diverse set of methods with different architectures for comparison. All models were pre-trained on NYU Depth V2 without fine-tuning their parameters. Among the models tested, Adabins [68] represents the depth estimation model using a pure Transformer architecture. Upon analyzing the results, we observed that our model exhibits

Table 2 Comparison of performances on the KITTI. The reported numbers are from the corresponding original papers. Measurements are made for the depth range from 0 m to 80 m. The best results are in **bold**, second best are underlined

Methods	$\delta_1 \uparrow$	$\delta_2 \uparrow$	$\delta_3 \uparrow$	AbsRel↓	SqRel↓	RMSE↓	$\text{RMSE}_{log} \downarrow$
Nekrasov *el al.*[55]	0.898	0.966	0.994	0.099	0.544	3.866	–
FastDepth[56]	0.839	0.966	0.990	0.139	0.714	4.952	0.132
FastDepth V2[56]	0.876	0.968	0.991	0.100	0.498	3.868	0.092
Ma *el al.*[59]	0.916	0.968	**0.997**	0.087	0.352	2.994	–
Amir *el al.*[62]	0.923	0.967	0.984	0.110	0.929	4.726	0.194
SGDepth [63]	0.879	0.961	0.981	0.113	0.835	4.693	–
An *el al.*[61]	0.915	0.985	0.996	0.087	0.347	3.107	0.084
MiniNet [64]	0.825	0.941	0.976	0.141	1.080	5.264	0.216
MonoFormer [65]	0.884	0.963	0.983	0.104	0.846	4.058	0.183
Lite-Mono [66]	0.886	0.963	0.983	0.107	0.765	4.561	0.183
Varma *el al.* [67]	0.851	0.952	0.980	0.125	0.905	5.096	0.203
EMTNet (Ours)	**0.928**	**0.988**	**0.997**	**0.082**	**0.324**	**2.946**	**0.075**

Table 3 Comparison of performances on SUN RGB-D test set without fine-tuning the models trained on NYU Depth V2. The best results are in **bold** and second best are <u>underlined</u>. The range of ground truth depth for evaluation from 0m to 8m

Methods	δ_1 ↑	δ_2 ↑	δ_3 ↑	AbsRel↓	RMSE↓	RMSE_{log} ↓
FastDepth[56]	0.669	0.744	0.812	0.188	0.691	0.279
MonoFormer (Hybrid)[65]	0.741	0.866	0.923	0.165	0.632	0.243
Lite-Mono[66]	0.717	0.854	0.910	0.168	0.656	0.247
AdaBins[68]	**0.771**	**0.944**	**0.983**	**0.159**	**0.476**	**0.211**
EMTNet (Ours)	<u>0.759</u>	<u>0.893</u>	<u>0.954</u>	<u>0.164</u>	<u>0.580</u>	<u>0.232</u>

Table 4 Quantitative comparison of different architecture models.

Methods	Architecture	AbsRel↓	#Params(M)↓	FPS↑
DepthNet Nano[58]	CNN	0.103	1.75	57
FastDepth[56]	CNN	0.139	3.96	53
MonoFormer (ViT) [65]	Transformer	0.118	54.2	27
MonoFormer (Hybrid)[65]	Hybrid	0.104	23.9	28
Lite-Mono[66]	Hybrid	0.107	10.3	41
AdaBins[68]	Transformer	0.058	78.0	21
DORN[31]	Transformer	0.072	99.8	18
EMTNet (Ours)	Hybrid	0.082	16.3	32

The test results used were performed using 224 × 224 images on a single NVIDIA GeForce RTX 2080 Ti graphics card. Hybrid in the table indicates the method using CNN and Transformer architectures

a slightly lower overall generalization ability compared to the Transformer architecture approach. However, when compared to hybrid architectures like MonoFormer [65] and Lite-Mono[66], our method demonstrates better generalization ability. Moreover, in the comparison with methods utilizing CNN architecture, our model emerges as the more advantageous choice in terms of generalization performance.

We conducted a comprehensive comparison of different models by evaluating both Params and real-time performance (FPS) on the same device, ensuring a fair assessment under identical conditions. Table 4 presents the test results, with AbsRel measured on the KITTI dataset. Theoretical analysis suggests that the Transformer architecture typically exhibits higher computational complexity than CNN-based models due to the inclusion of the attention module. The test results show that our model showcases a significant advantage, boasting fewer parameters compared to a model employing the Transformer architecture. Furthermore, when pitted against hybrid architectures such as Lite-Mono[66] and MonoFormer (Hybrid)[65], our model outperforms in terms of accuracy. In regard to real-time performance, our model is slower than pure CNN-based models but faster and more accurate than pure Transformer-based models. This trade-off allows our approach to strike an optimal balance

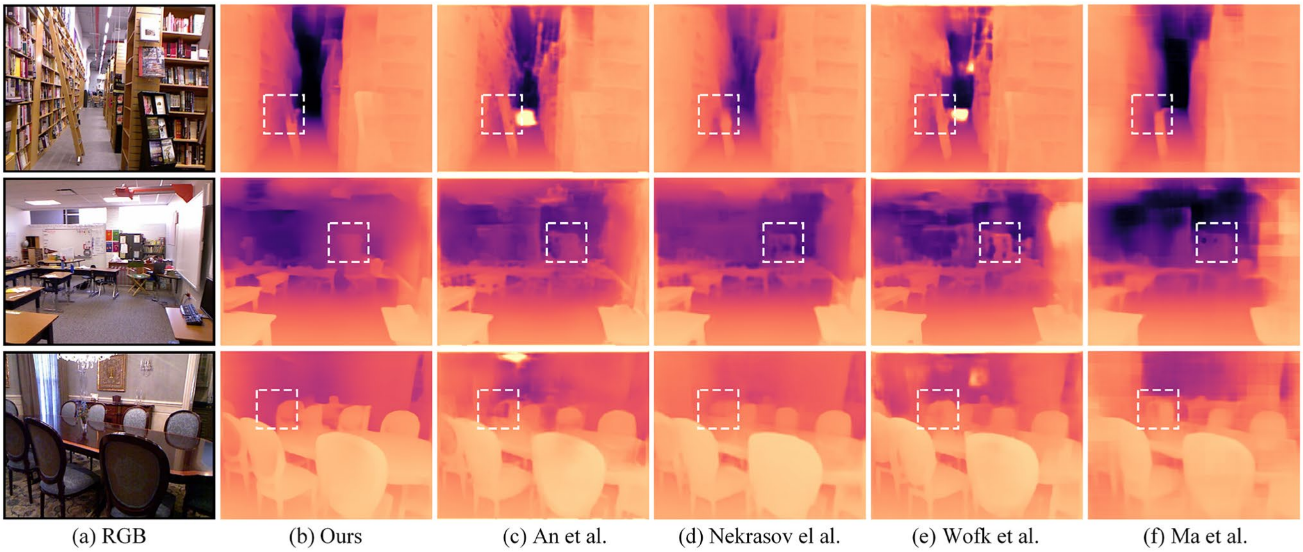

Fig. 4 Qualitative comparison with An et al. [61], Nekrasov et al. [55], Wofk et al. [56], Ma et al. [59]. All the models are pre-trained on NYU Depth V2 [22] training set

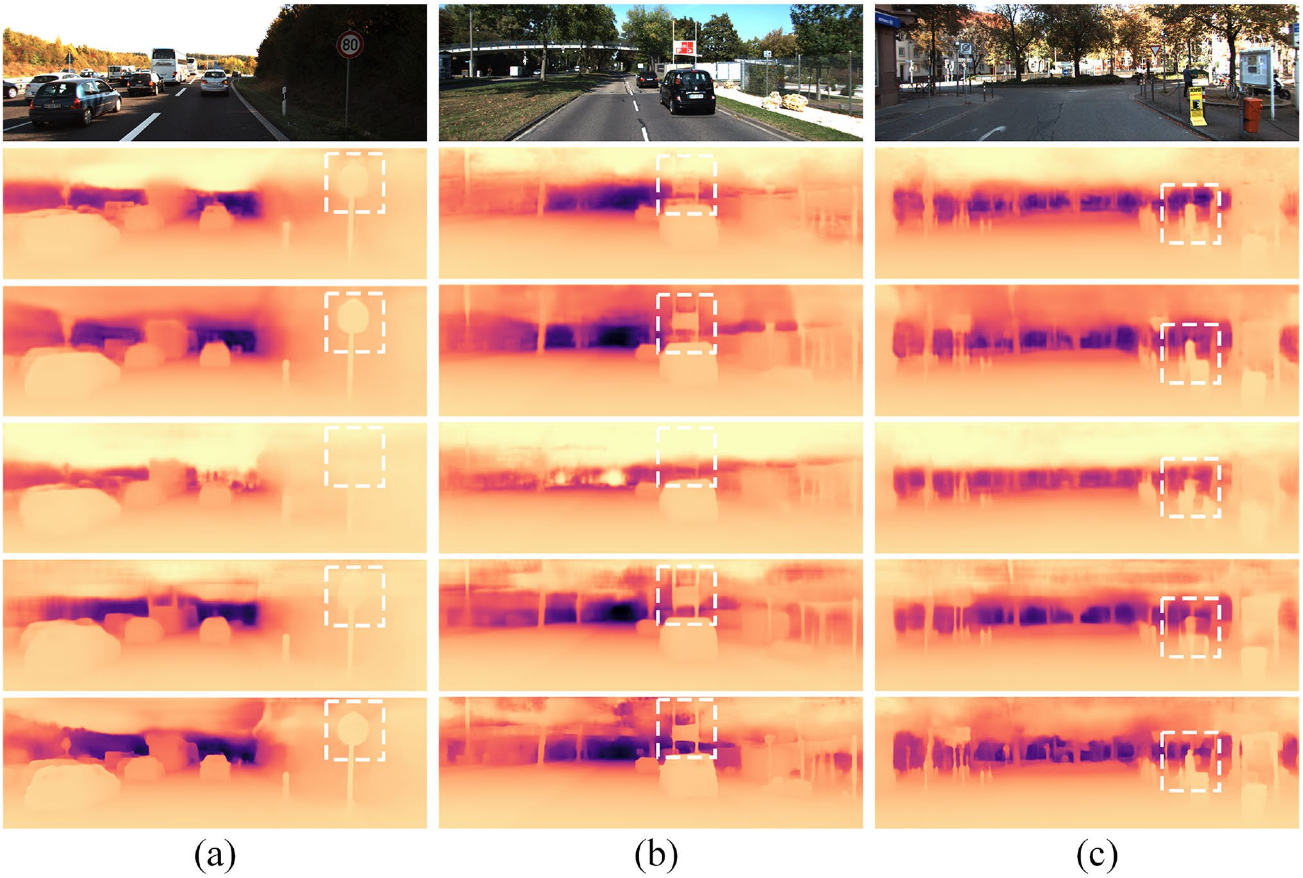

Fig. 5 Qualitative comparison on KITTI Eigen split [23]. For each column, from top to bottom we present the input image, the prediction from An et al. [61], Nekrasov et al. [55], Wofk et al. [56], Ma et al. [59] and EMTNet (ours)

concerning the number of parameters and computational efficiency within the hybrid architecture models. As a result, our model achieves remarkable results in terms of both accuracy and computational performance.

4.4 Ablation study

In the ablation study, we evaluated the impact that the following different design choices had on our model.

Depthwise separable convolution (DSC). DSC originates from MobileNet [69], a lightweight model for computer vision tasks that is often used in environments with limited hardware resources. Intuitively, its use allows our model to be more adaptable to real-time tasks. Therefore, during the experimental stage, we used DSC in the skip-connection part between the encoder and the decoder. However, the experiments showed that including DSC reduced the number of parameters in the model, but the model's accuracy decreased, which was not desirable.

Activation function. In our experiments, we used both ReLU and GeLU activation functions to verify the real-time performance of the model. ReLU is a linear function, while GeLU is nonlinear in the real number domain. In theory, using linear activation functions in environments with limited hardware resources allows for faster information processing. However, our experiments showed that using GeLU was superior, which was surprising.

MTB Block. MTB is the module we introduced to reduce the number of model parameters and FLOPs. In the ablation experiments, we compared the outcomes of using MTB and traditional self-attention and found that using MTB led to better performance. Although it is not as good in real-time as using DSC, it showed better accuracy performance (Table 5).

4.5 Experimental discussion

We propose a real-time monocular depth estimation model named EMTNet, which is built upon the Mobile Transformer Block (MTB). EMTNet effectively integrates CNN and ViT, enabling the extraction of both local and global features in complex scenes. This synergy accounts for the network's capacity to enhance depth map details and exhibit robust generalization capabilities. Furthermore, the Branch Sharing

Table 5 Ablation results of the DSC, ReLU, GeLU and MTB.

DSC	ReLU	GeLU	MTB	AbsRel	RMSE	FPS
✓		✓	✓	0.174	0.586	36.93
	✓		✓	0.141	0.490	31.74
		✓	✓	**0.141**	**0.490**	**32.55**
	✓			0.192	0.651	27.77
		✓		0.192	0.651	27.69

The best results for the combined performance are shown in **bold** font

scheme employed by MTB efficiently reduces the model's parameter, thereby endowing it with the capability for real-time depth estimation. EMTNet is going to outperform the other models in terms of the detail performance of the output depth map. However, using MTB comes with a trade-off between accuracy and real-time performance. While the model's accuracy is not significantly improved compared to previous works on the NYU Depth V2 dataset, its real-time efficiency is compromised. In the generalization test on the SUN RGB-D dataset, our network performs well with the same hybrid architecture, but there is still a lot of room for improvement compared to the network using the pure Transformer architecture.

Our aim is to enhance the model's accuracy and real-time performance, but using a global attention approach could cause a decline in real-time performance. Therefore, striking a balance between real-time and accuracy is challenging for depth estimation models. Furthermore, high-resolution depth maps are not always necessary for most depth estimation tasks, as depth information is often correlated with continuity over most regions. High-resolution outputs could only make sense in complex depth scenes, but they also reduce the model's real-time performance. Also the high-resolution output reduces the real-time nature of the model. Thus far, we have achieved our desired results in terms of model accuracy and real-time performance.

5 Limitations and future directions

Our method has demonstrated promising results in experiments conducted on two datasets, surpassing CNN-based models and even some Transformer-based methods. However, we acknowledge that our model still has certain limitations due to architectural design deficiencies and model training issues. One major concern is the substantial number of parameters and computational complexity of our model compared to the other hybrid model (CNN+Transformer). Redundant computations also pose a challenge. Moreover, we observed variations in prediction accuracy across different datasets, with the model performing less effectively on NYU Depth V2 compared to KITTI. Some metrics showed only marginal improvement, and in some cases, even a reduction was observed. We suspect that the lack of coordination between the CNN and Transformer components during the depth feature extraction in the encoder stage results in the loss of important features during transmission. Additionally, our model faced convergence issues during training, necessitating the setting of multiple epochs for slow convergence.

To address these limitations and enhance our method, we are planning to explore alternative advanced modeling approaches. This exploration could involve delving into a pure Transformer architecture paradigm with pre-trained parameter initialization, along with an investigation into the integration of a depth-interval categorization (Bins-based) methodology to expedite the model's convergence speed. Additionally, within the model's training regimen, we have deliberately incorporated a broader array of data augmentation techniques. This strategic augmentation of the training dataset contributes significantly to amplifying the model's generalization prowess. Moreover, we aspire to examine the model's adaptability and the potential application of its enhanced methodologies in a wide range of visual tasks. These tasks encompass, but are not limited to, semantic segmentation, target detection, and multi-image 3D reconstruction.

6 Conclusion

We present EMTNet, an innovative real-time monocular depth estimation model constructed upon the Mobile Transformer Block (MTB). This model synergistically harnesses the capabilities of both CNN and ViT architectures to elevate feature extraction across local and global domains. Leveraging the Branch Sharing scheme within MTB, EMTNet successfully achieves parameter reduction, thereby optimizing its aptitude for real-time depth estimation tasks. To produce finer depth maps, we synthesize high-resolution depth maps by fusing multi-scale features in the decoder section. Our model achieves good results on two benchmark datasets. When comparing the output prediction maps, our model demonstrates superior ability in generating high-quality depth maps, especially in complex scenes. Moreover, in depth missing regions, our model excels in depth completion compared to other models. In terms of real-time performance, our approach achieves 32 frames per second, striking a harmonious equilibrium between accuracy and speed.

References

1. Diaz, C., Walker, M., Szafir, D. A., and Szafir, D. (2017) Designing for depth perceptions in augmented reality. In: 2017 IEEE international symposium on mixed and augmented reality (ISMAR), pages 111–122. IEEE
2. Kusupati, U., Cheng, S., Chen, R., and Su, H. (2020) Normal assisted stereo depth estimation. In: Proceedings of the IEEE/cvf conference on computer vision and pattern recognition, pp 2189–2199
3. Mancini M, Costante G, Valigi P, Ciarfuglia TA (2016) Fast robust monocular depth estimation for obstacle detection with fully convolutional networks. In: 2016 IEEE/rsj international conference on intelligent robots and systems (IROS), pp 4296–4303. IEEE
4. Mur-Artal R, Montiel JMM, Tardós JD (2015) Orb-slam: a versatile and accurate monocular slam system. IEEE Trans Rob 31(5):1147–1163
5. Ha H, Im S, Park J, Jeon H-G, Kweon IS (2016) High-quality depth from uncalibrated small motion clip. In: Proceedings of the IEEE conference on computer vision and pattern Recognition, pp 5413–5421
6. Kong N, Black MJ (2015) Intrinsic depth: improving depth transfer with intrinsic images. In: Proceedings of the IEEE international conference on computer vision, pp 3514–3522
7. Karsch K, Liu C, Kang SB (2016) Depth transfer: depth extraction from videos using nonparametric sampling. In: dense image correspondences for computer vision, pp 173–205. Springer
8. Rajagopalan AN, Chaudhuri S, Mudenagudi U (2004) Depth estimation and image restoration using defocused stereo pairs. IEEE Trans Pattern Anal Mach Intell 26(11):1521–1525
9. Eigen D, Fergus R (2015) Predicting depth, surface normals and semantic labels with a common multi-scale convolutional architecture. In: Proceedings of the IEEE international conference on computer vision, pp 2650–2658
10. Liu F, Shen C, Lin G (2015) Deep convolutional neural fields for depth estimation from a single image. In: Proceedings of the IEEE conference on computer vision and pattern recognition, pp 5162–5170
11. Porzi L, Bulo SR, Penate-Sanchez A, Ricci E, Moreno-Noguer F (2016) Learning depth-aware deep representations for robotic perception. IEEE Robotics and Autom Lett 2(2):468–475
12. Vaswani A, Shazeer N, Parmar N, Uszkoreit J, Jones L, Gomez AN, Kaiser Łukasz, Polosukhin I (2017) Attention is all you need. Advances in neural information processing systems, pp 30
13. Dosovitskiy A, Beyer L, Kolesnikov A, Weissenborn D, Zhai X, Unterthiner T, Dehghani M, Minderer M, Heigold G, Gelly S, et al (2020) An image is worth 16x16 words: transformers for image recognition at scale. arXiv preprint arXiv:2010.11929
14. Bhat SF, Alhashim I, Wonka P (2021) Adabins: Depth estimation using adaptive bins. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, pp 4009–4018
15. Li Z, Wang X, Liu X, Jiang J (2022) Binsformer: revisiting adaptive bins for monocular depth estimation. arXiv preprint arXiv:2204.00987
16. Zhao C, Zhang Y, Poggi M, Tosi F, Guo X, Zhu Z, Huang G, Tang Y, Mattoccia S (2022) Monovit: self-supervised monocular depth estimation with a vision transformer. arXiv preprint arXiv:2208.03543
17. Bae J, Moon S, Im S (2022) Deep digging into the generalization of self-supervised monocular depth estimation. arXiv preprint arXiv:2205.11083
18. Li Z, Chen Z, Liu X, Jiang J (2022) Depthformer: exploiting long-range correlation and local information for accurate monocular depth estimation. arXiv preprint arXiv:2203.14211
19. Shu C, Chen Z, Chen L, Ma K, Wang M, Ren H (2022) Sidert: A real-time pure transformer architecture for single image depth estimation. arXiv preprint arXiv:2204.13892
20. Ma H, Xia X, Wang X, Xiao X, Li J, Zheng M (2022) MoCoViT: mobile convolutional vision transformer
21. Ranftl R, Bochkovskiy A, Koltun V (2021) Vision transformers for dense prediction. In: Proceedings of the IEEE/CVF International Conference on Computer Vision, pages 12179–12188
22. Silberman N, Hoiem D, Kohli P, Fergus R (2012) Indoor segmentation and support inference from rgbd images. In: European conference on computer vision, pp 746–760. Springer
23. Geiger A, Lenz P, Stiller C, Urtasun R (2013) Vision meets robotics: the kitti dataset. Int J Robotics Res 32(11):1231–1237
24. Saxena A, Chung S, Ng A (2005) Learning depth from single monocular images. Advances in neural information processing systems, 18
25. Karsch K, Liu C, Kang SB (2019) Depth extraction from video using non-parametric sampling. arXiv preprint arXiv:2002.04479
26. Konrad J, Wang M, Ishwar P (2012) 2d-to-3d image conversion by learning depth from examples. In: 2012 IEEE Computer society conference on computer vision and pattern recognition workshops, pp 16–22. IEEE
27. Karsch K, Liu C, Kang SB (2014) Depth transfer: depth extraction from video using non-parametric sampling. IEEE Trans Pattern Anal Mach Intell 36(11):2144–2158
28. Liu M, Salzmann M, He X (2014) Discrete-continuous depth estimation from a single image. In: Proceedings of the IEEE conference on computer vision and pattern recognition, pp 716–723
29. Eigen D, Puhrsch C, Fergus R (2014) Depth map prediction from a single image using a multi-scale deep network. Advances in neural information processing systems, 27
30. Laina I, Rupprecht C, Belagiannis V, Tombari F, Navab N (2016) Deeper depth prediction with fully convolutional residual networks. In: 2016 Fourth international conference on 3D vision (3DV), pp 239–248. IEEE
31. Fu H, Gong M, Wang C, Batmanghelich K, Tao D (2018) Deep ordinal regression network for monocular depth estimation. In: Proceedings of the IEEE conference on computer vision and pattern recognition, pp 2002–2011
32. Liu F, Shen C, Lin G, Reid I (2015) Learning depth from single monocular images using deep convolutional neural fields. IEEE Trans Pattern Anal Mach Intell 38(10):2024–2039
33. He K, Zhang X, Ren S, Sun J (2016) Deep residual learning for image recognition. 2016 IEEE conference on computer vision and pattern recognition (CVPR), pp 770–778
34. Qi X, Liao R, Liu Z, Urtasun R, Jia J (2018) Geonet: geometric neural network for joint depth and surface normal estimation. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp 283–291
35. Touvron H, Cord M, Douze M, Massa F, Sablayrolles A, Jégou H (2021) Training data-efficient image transformers & distillation through attention. In: International conference on machine learning, pp 10347–10357. PMLR
36. Wang W, Xie E, Li X, Fan D-P, Song K, Liang D, Lu T, Luo P, Shao L (2021) Pyramid vision transformer: a versatile backbone for dense prediction without convolutions. In: Proceedings of

the IEEE/CVF international conference on computer vision, pp 568–578
37. Han K, Xiao A, Enhua W, Guo J, Chunjing X, Wang Y (2021) Transformer in transformer. Adv Neural Inf Process Syst 34:15908–15919
38. Xu W, Xu Y, Chang T, Tu Z (2021) Co-scale conv-attentional image transformers. In: Proceedings of the IEEE/CVF international conference on computer vision, pp 9981–9990
39. Liu Z, Lin Y, Cao Y, Hu H, Wei Y, Zhang Z, Lin S, Guo B (2021) Swin transformer: hierarchical vision transformer using shifted windows. In: Proceedings of the IEEE/CVF international conference on computer vision, pp 10012–10022
40. Dalmaz O, Yurt M, Çukur T (2021) Resvit: residual vision transformers for multimodal medical image synthesis. IEEE Trans Med Imaging 41:2598–2614
41. Mehta S, Rastegari M (2021) Mobilevit: Light-weight, general-purpose, and mobile-friendly vision transformer. arXiv:2110.02178
42. Li Z, Li Y, Li Q, Zhang Y, Wang P, Guo D, Lu L, Jin D, Hong Q (2022) Lvit: Language meets vision transformer in medical image segmentation. arXiv:2206.14718
43. Wu K, Zhang J, Peng H, Liu M, Xiao B, Fu J, Yuan L (2022) Tinyvit: fast pretraining distillation for small vision transformers. arXiv:2207.10666
44. Dai Z, Liu H, Le QV, Tan M (2021) Coatnet: marrying convolution and attention for all data sizes. Adv Neural Inf Process Syst 34:3965–3977
45. Srinivas A, Lin T-Y, Parmar N, Shlens J, Abbeel P, Vaswani A (2021) Bottleneck transformers for visual recognition. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, pp 16519–16529
46. d'Ascoli S, Touvron H, Leavitt ML, Morcos AS, Biroli G, Sagun L (2021) Convit: improving vision transformers with soft convolutional inductive biases. In: International Conference on Machine Learning, pp 2286–2296. PMLR
47. Wu H, Xiao B, Codella N, Liu M, Dai X, Yuan L, Zhang L (2021) Cvt: introducing convolutions to vision transformers. In: Proceedings of the IEEE/CVF International Conference on Computer Vision, pp 22–31
48. Graham B, El-Nouby A, Touvron H, Stock P, Joulin A, Jégou H, Douze M (2021) Levit: a vision transformer in convnet's clothing for faster inference. In: Proceedings of the IEEE/CVF international conference on computer vision, pp 12259–12269
49. LeCun Y, Boser B, Denker J, Henderson D, Howard R, Hubbard W, Jackel L (1989) Handwritten digit recognition with a back-propagation network. Advances in neural information processing systems, 2
50. Han K, Wang Y, Tian Q, Guo J, Xu C, Xu C (2020) Ghostnet: more features from cheap operations. In: 2020 IEEE/CVF conference on computer vision and pattern recognition (CVPR), pp 1577–1586
51. Hu J, Shen L, Sun G (2018) Squeeze-and-excitation networks. In: Proceedings of the IEEE conference on computer vision and pattern recognition, pp 7132–7141
52. Lee JH, Han M-K, Ko DW, Suh IH (2019) From big to small: multi-scale local planar guidance for monocular depth estimation. arXiv preprint arXiv:1907.10326
53. Everingham M, Van Gool L, Williams CKI, Winn J, Zisserman A (2010) The pascal visual object classes (voc) challenge. Int J Comput Vision 88(2):303–338
54. Smith LN, Nicholay T (2019) Super-convergence: very fast training of neural networks using large learning rates. In: Tien Pham (ed) Artificial Intelligence and Machine Learning for Multi-Domain Operations Applications. International Society for Optics and Photonics, SPIE, vol 11006, pp 1100612
55. Nekrasov V, Dharmasiri T, Spek A, Drummond T, Shen C, Reid ID (2018) Real-time joint semantic segmentation and depth estimation using asymmetric annotations. In: 2019 international conference on robotics and automation (ICRA), pp 7101–7107
56. Wofk D, Ma F, Yang T-J, Karaman S, Sze V (2019) Fastdepth: fast monocular depth estimation on embedded systems. In: 2019 international conference on robotics and automation (ICRA), IEEE, pp 6101–6108.
57. Spek A, Dharmasiri T, Drummond T (2018) Cream: condensed real-time models for depth prediction using convolutional neural networks. 2018 IEEE/RSJ international conference on intelligent robots and systems (IROS), pp 540–547
58. Wang L, Famouri M, Wong A (2020) Depthnet nano: a highly compact self-normalizing neural network for monocular depth estimation. arXiv:2004.08008
59. Ma F, Karaman S (2017) Sparse-to-dense: depth prediction from sparse depth samples and a single image. 2018 IEEE international conference on robotics and automation (ICRA), pp 1–8
60. Yucel MK, Dimaridou V, Drosou A, Saà-Garriga A (2021) Real-time monocular depth estimation with sparse supervision on mobile. 2021 IEEE/CVF conference on computer vision and pattern recognition workshops (CVPRW), pp 2428–2437
61. An S, Zhou F, Yang M, Zhu H, Fu C, Tsintotas KA (2021) Real-time monocular human depth estimation and segmentation on embedded systems. 2021 IEEE/RSJ international conference on intelligent robots and systems (IROS), pp 55–62
62. Atapour-Abarghouei A, Breckon TP (2018) Real-time monocular depth estimation using synthetic data with domain adaptation via image style transfer. In: 2018 IEEE/CVF conference on computer vision and pattern recognition, pp 2800–2810
63. Klingner M, Termöhlen J-A, Mikolajczyk J, Fingscheidt T (2020) Self-supervised monocular depth estimation: solving the dynamic object problem by semantic guidance. In: european conference on computer vision
64. Liu J, Li Q, Cao R, Tang W, Qiu G (2020) MiniNet: an extremely lightweight convolutional neural network for real-time unsupervised monocular depth estimation. ISPRS J Photogrammetry Remote Sens 166:255–267
65. Bae J-H, Moon S, Im S (2022) Deep digging into the generalization of self-supervised monocular depth estimation. Proceedings of the AAAI conference on artificial intelligence
66. Zhang N, Nex F, Vosselman G, Kerle N (2022) Lite-mono: a lightweight cnn and transformer architecture for self-supervised monocular depth estimation. arXiv:2211.13202
67. Varma A, Chawla H, Zonooz B, Arani E (2022) Transformers in self-supervised monocular depth estimation with unknown camera intrinsics. arXiv:2202.03131
68. Bhat SF, Alhashim I, Wonka P (2021) Adabins: depth estimation using adaptive bins. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, pp 4009–4018
69. Howard AG, Zhu M, Chen B, Kalenichenko D, Wang W, Weyand T, Andreetto M, Adam H (2017) Mobilenets: Efficient convolutional neural networks for mobile vision applications. arXiv:1704.04861