Exploration for Lightweight Monocular Depth Estimation

Wenxuan Zhang

November 28, 2023

Abstract

Monocular depth estimation, a key challenge in computer vision, has profound implications for fields like augmented reality, autonomous driving, and robotics. Despite the significant advancements with deep learning, the practical deployment of these models is often hindered by their computational cost, memory requirements, and slower inference speeds. This paper explores the pursuit of lightweight models that balance computational efficiency with accurate depth estimation. We delve into various strategies, including network compression techniques, efficient network architectures, multi-task learning, and inference optimization. Although promising, the exploration of lightweight monocular depth estimation presents numerous challenges, opening avenues for future research geared towards models that are accurate, efficient, and practical for real-world applications.

1 Introduction

Monocular depth estimation, a key task in the field of computer vision, revolves around estimating the distance from a camera to various points in a scene using a single image. The implications of accurately solving this problem are immense, with potential applications spanning across augmented reality, autonomous driving, robotics, and more. However, deploying these depth estimation models in practical scenarios is often a challenge due to issues such as high computational cost, large memory requirements, and slower inference speeds.

Over the years, depth estimation techniques have evolved significantly. Traditional methods relied on hand-crafted features and stereo images to estimate depth. However, the emergence of deep learning techniques, particularly convolutional neural networks (CNNs), has brought about a paradigm shift in the field. CNNs, with their ability to learn predictive features directly from data, have enhanced the performance of depth estimation, particularly from single images.

Figure 1: An illustration for the target of monocular depth estimation: the left image is the input RGB photo of the scene to be estimated, the middle is the model output of depth estimation, and the right is the ground truth.

Despite these advancements, the deployment of depth estimation models in real-world applications is not straightforward. The key challenges include high computational cost, large memory requirements, and slower inference speeds. These issues particularly become prominent when deploying these models on edge devices with limited computational resources and power. Therefore, there is an urgent need for models that can maintain a balance between computational efficiency and accurate depth estimation.

To address the above challenges, the focus has shifted towards lightweight models. These models are designed to provide accurate depth estima-

tion without consuming excessive computational resources, thereby presenting an important trade-off between model performance and computational efficiency. This balance is of paramount importance in real-world systems where resources are often limited.

The pursuit for lightweight models has led researchers to explore various strategies. One such strategy includes network compression techniques such as pruning, quantization, and knowledge distillation. These techniques aim to reduce the size of the model while maintaining its performance. Another strategy involves designing efficient network architectures like MobileNet and EfficientNet that use depthwise separable convolutions and scaling techniques to reduce computational demands. Multi-task learning is another approach where a model is trained to perform multiple related tasks simultaneously, thereby exploiting the common features among these tasks. Finally, inference optimization techniques such as hardware-aware neural architecture search are being explored to design models that are optimized for specific hardware platforms, thereby improving inference time.

The journey towards lightweight monocular depth estimation models is a fascinating one, with a constant balancing act of maintaining efficiency while ensuring performance. The advancements in this field are promising. However, they also present an array of challenges and opportunities for future research. The ultimate goal is clear: to develop models that are not only accurate and efficient but also practical for deployment in real-world applications.

2 Problem Description

2.1 Background

Monocular Depth Estimation (MDE) refers to the process of predicting the depth information from a single image. This task, although seemingly straightforward to human vision due to our innate perception abilities, poses a daunting challenge for computer vision systems. The reason being the inherent ambiguity in mapping 2D image pixels to 3D world points, a problem known as the scale ambiguity problem. While humans can easily infer depth from visual cues, such as size, perspective, and texture gradient, automated systems struggle with these tasks.

The traditional approaches to depth estimation relied on stereo vision, where depth information was inferred from the disparity between two views of the same scene. However, these methods require a stereo camera setup, which increases hardware requirements and complexity. Furthermore, they are prone to errors in textureless or occluded regions.

The advent of deep learning has revolutionized depth estimation by allowing models to learn complex mappings from color images to depth maps. While these methods have achieved significant improvements over traditional methods, they have their own set of challenges. The primary one is the requirement of large amounts of ground-truth depth data for training. Collecting such data is a challenging and time-consuming process, often requiring specialized equipment like LiDAR.

Another significant challenge is the deployment of these models in real-world applications. Most of the state-of-the-art models have millions of parameters, leading to high computational costs, large memory requirements, and slower inference speeds. This hinders their effective deployment on edge devices that have limited computational resources and power. Hence, the practical application of these models demands a careful balance between accuracy and efficiency.

The problem, thus, is to develop lightweight, yet accurate, models for monocular depth estimation. These models should be capable of running on devices with limited resources, making them suitable for real-world applications like autonomous driving, augmented reality, and robotics. Balancing this trade-off between accuracy and efficiency, while also handling the inherent ambiguity in monocular depth estimation, constitutes the crux of this problem. The exploration of potential solutions for this problem presents a plethora of opportunities for future research in the field of computer vision.

2.2 Lite-weight MDE

Lightweight MDE models are the proposed solution to this problem. However, designing these models is a complex task due to several reasons. First, reducing the size of the model often comes at the cost of model performance. Simplifying the model architecture might lead to faster inference times and lower memory requirements, but it may also result in a significant drop in accuracy. Therefore, the primary challenge is to strike a balance between model complexity and accuracy.

Second, traditional methods of network compression such as pruning, quantization, and knowledge distillation, while effective in reducing the size of the model, have to be carefully applied to avoid significant degradation in depth estimation performance.

Moreover, the design of efficient network architectures, such as MobileNet and EfficientNet, brings its own set of challenges. These architectures use depthwise separable convolutions and scaling techniques to reduce computational demands, but tailoring these architectures for the specific task of MDE is a non-trivial task.

Another challenge is the incorporation of multi-task learning, which trains a model to perform multiple related tasks simultaneously. While this approach can improve the model's performance by exploiting common features among tasks, it adds another layer of complexity in terms of model design and training.

Lastly, hardware-aware neural architecture search techniques, which design models optimized for specific hardware platforms, are still an active area of research and pose their own set of challenges.

In summary, the problem lies in developing lightweight MDE models that maintain a high level of accuracy. The design of these models should consider factors such as model complexity, the application of network compression techniques, the use of efficient network architectures, the integration of multi-task learning, and the optimization for specific hardware platforms. The exploration and resolution of these challenges represent a significant opportunity for advancing MDE and its real-world applications.

2.3 Formal Definition

Monocular Depth Estimation (MDE) is a task where a model is expected to learn a mapping function from a set of input images to their corresponding depth maps. Let's denote the set of input images as $I = \{I_1, I_2, ..., I_n\}$ and their corresponding ground truth depth maps as $D = \{D_1, D_2, ..., D_n\}$. The goal is to find a function $f : I \rightarrow D$ such that the predicted depth map for an image I_i is as close as possible to the ground truth depth map D_i.

However, traditional MDE models, which leverage deep learning techniques, often have a high computational cost and large memory requirements. This can be quantitatively measured using parameters such as the number of floating point operations per second (FLOPs) and the size of the model (in MB). Let's denote the computational cost and size of a model M as $C(M)$ and $S(M)$ respectively. For a device with computational capacity C_{max} and memory S_{max}, a traditional MDE model M_t often leads to $C(M_t) > C_{max}$ and $S(M_t) > S_{max}$, thereby making it unsuitable for deployment on such devices.

In contrast, the goal of a lightweight MDE model is to achieve a balance between accuracy and computational efficiency. Formally, given a set of input images I and their corresponding depth maps D, the goal is to find a function $f : I \rightarrow D$ such that the mean absolute error (MAE) between the predicted and ground truth depth maps is minimized, while also ensuring that the computational cost $C(M)$ and size $S(M)$ of the model M are within the allowable limits of the device. Mathematically, this can be represented as:

$$\begin{aligned} \underset{M}{\text{minimize}} \quad & \frac{1}{n}\sum_{i=1}^{n} ||D_i - f(I_i)||_1 \\ \text{subject to} \quad & C(M) \leq C_{max}, \\ & S(M) \leq S_{max}, \end{aligned} \tag{1}$$

where $||.||_1$ represents the L1 norm or MAE, and n is the total number of images.

The primary challenge lies in finding the optimal model M that satisfies these constraints. It requires exploring different strategies, such as network compression techniques, efficient network architectures,

multi-task learning, and hardware-aware optimization, which add to the complexity of the problem.

2.4 Datasets

The KITTI dataset [4], a cornerstone in the realm of computer vision and autonomous driving research, was birthed from a collaboration between the Karlsruhe Institute of Technology and the Toyota Technological Institute at Chicago. This rich dataset offers a comprehensive suite of benchmarks aimed at various tasks pivotal for the progression of autonomous vehicles.

Data collection took place in the picturesque city of Karlsruhe, Germany. A station wagon equipped with cutting-edge sensors—including a Velodyne LiDAR and a mix of grayscale and color cameras—roamed through urban streets, highways, and rural sequences, capturing a diverse array of scenarios. This data forms the foundation upon which researchers worldwide have benchmarked their algorithms, especially those tailored for the challenges of autonomous driving.

The breadth of the KITTI dataset is what truly sets it apart. It serves as a testing ground for a plethora of tasks. From stereo vision, which assists in evaluating stereo and optical flow algorithms, to depth evaluation, which aids both monocular and stereo depth estimation methods—the KITTI dataset has it all. Moreover, its meticulously curated annotations for 3D object detection, 2D object tracking, road/lane detection, and even semantic segmentation underscore its comprehensive nature.

However, like all benchmarks, KITTI is not without its limitations. A significant challenge arises from its specific environmental bias. Given its focus on German streets and surroundings, there's a palpable geographical constraint. Models that are solely trained on the KITTI dataset might grapple with generalization when faced with diverse driving terrains from different corners of the globe.

In MDE realm, the KITTI dataset is also of great importance. Monocular Depth Estimation (MDE), the intricate task of discerning depth from a singular image, finds its proving ground in real-world datasets. Among these, the KITTI dataset stands out prominently. Originating from the streets of Karlsruhe, Germany, KITTI serves as a critical benchmark for a myriad of computer vision tasks, especially MDE.

What makes KITTI invaluable for MDE is its treasure trove of diverse driving scenarios. From bustling urban streets to tranquil rural settings and fast-paced highways, the dataset captures the multifaceted realities of driving. Researchers, using KITTI, are not merely testing their algorithms in sterile, synthetic environments but in dynamic, real-world situations. This realism, combined with the dataset's LiDAR-based depth maps, sets a rigorous standard for depth estimation models. With these depth maps acting as ground truth, the dataset challenges and validates the prowess of MDE algorithms.

Moreover, the multi-modal nature of KITTI's data, encompassing both visual imagery and LiDAR readings, brings forth another layer of depth (pun intended) to the testing environment. It allows a direct juxtaposition between predicted depth from monocular images and actual depth readings, enabling researchers to fine-tune and rectify their models with unparalleled precision. Additionally, the competitive spirit fostered by the KITTI leaderboard pushes researchers to continuously innovate, ensuring that the MDE field remains in perpetual motion.

Yet, while KITTI's virtues are many, it is not without its limitations. The dataset, deeply rooted in the specific landscapes and environments of Germany, could inadvertently introduce geographical biases. Models honed and perfected on KITTI might falter when faced with terrains or driving nuances alien to the dataset's confines.

To encapsulate, the KITTI dataset, with its robust and real-world data, has profoundly influenced the trajectory of Monocular Depth Estimation research. It has been both a challenge and a validator, a beacon guiding the evolution of MDE. However, as the field matures, there's a growing emphasis on diversifying datasets, ensuring that depth estimation models are as versatile as they are accurate.

3 Methodologies and Examples of Related Work

As we venture deeper into the realm of Monocular Depth Estimation (MDE), we encounter a unique challenge: how do we strike a balance between model accuracy and computational efficiency? This question takes us on an exciting journey through various innovative strategies. Each strategy is like a path leading us through the dense forest of model complexity, guiding us towards the ultimate goal: lightweight MDE models that maintain high accuracy. Let's embark on this journey and explore the four primary paths.

3.1 Network Compression Techniques

Network compression techniques aim to reduce the computational complexity and size of neural networks, without significantly compromising the performance. Two popular techniques within this approach are network pruning and knowledge distillation.

Network Pruning

Network pruning involves eliminating unimportant or redundant connections within a neural network. The idea is that not all connections in a neural network contribute to the final output, and some of these can be pruned away without affecting the overall performance significantly. A pioneering work in this field is by Han et al., who demonstrated that large neural networks often contain redundancy in the form of unimportant connections and weights [5].

However, conventional pruning techniques could lead to irregular network structures that are not hardware-friendly. To overcome this, He et al. proposed a structured pruning method which prunes entire channels of convolutional layers to maintain the benefits of hardware acceleration [6].

Knowledge Distillation

Knowledge distillation is another network compression technique where a smaller network (student) is trained to mimic the behavior of a larger network (teacher). This technique is based on the observation that the output of a neural network includes both the final prediction as well as the knowledge embedded in the network's structure and learned parameters. Hinton et al. introduced this technique, showing that a smaller network could achieve comparable performance to a larger one if it's trained to mimic the soft outputs (logits) of the larger network [7].

This approach has been further refined by Romero et al., where the student is trained to mimic intermediate representations (hints) of the teacher network, enabling the student to learn a richer function [21].

In conclusion, network compression techniques, mainly through network pruning and knowledge distillation, play a crucial role in reducing the size of neural networks and the computational resources needed, making it possible to deploy these networks on resource-constrained devices without a significant loss in performance.

3.2 Efficient Network Architectures

This primarily involves designing a network architecture that works effectively in theory and practice. For instance, MobileNet introduces an efficient convolution method called depthwise separable convolution, which significantly reduces computation and model size. Another network, EfficientNet, creates an efficient network architecture by balancing the scaling of network depth, width, and resolution.

Efficient network architectures are designed with the primary objective of being lightweight while still delivering robust performance. Two of the most prominent examples of this approach are MobileNet and EfficientNet.

MobileNet

MobileNet is a family of neural network architectures designed with the aim of being lightweight and efficient. The key to MobileNet's efficiency is its use of depthwise separable convolutions, a type of convolution that significantly reduces the computational cost and model size without a major impact on accuracy. Depthwise separable convolution splits the standard convolution operation into a depthwise convolution and a pointwise convolution, reducing both the computational cost and the number of parameters. This makes MobileNet architectures particularly suited for mobile devices and other environments where computational resources and memory

are limited [8].

EfficientNet

EfficientNet, proposed by Mingxing Tan and Quoc V. Le, is another family of efficient network architectures. The central idea behind EfficientNet is the use of a compound scaling method to balance the scaling of network depth, width, and resolution. While previous works mainly focused on scaling these factors separately, EfficientNet balances all three based on a compound coefficient. This approach leads to better performance as it considers the interplay between different scaling dimensions. The authors found that using their compound scaling method, they could achieve state-of-the-art accuracy with significantly fewer parameters and computational cost [24].

Both MobileNet and EfficientNet have set a strong foundation in the pursuit of developing efficient neural network architectures. Their principles and techniques have been widely adopted and improved upon in subsequent research, contributing significantly to the advancement of lightweight and efficient deep learning models, especially for resource-constrained devices.

3.3 Hardware-aware Neural Architecture Search (NAS)

To obtain models with efficient inference performance, researchers are starting to consider hardware constraints. They use Neural Architecture Search (NAS) techniques to automatically find models that are best suited to specific hardware platforms (e.g., mobile phones or embedded devices). These models offer faster inference speeds and lower energy consumption on specific hardware platforms.

This method relies on advances in hardware technology to speed up computation, rather than modifying the models themselves. Two key approaches in this category are the use of Graphic Processing Units (GPUs) and Application-Specific Integrated Circuits (ASICs), such as the Tensor Processing Units (TPUs) developed by Google.

GPUs

Originally designed for rendering video games and computer graphics, GPUs have proven to be very effective for parallel computations, making them a popular choice for training and deploying deep learning models. GPUs have a large number of cores, which allows them to perform many computations simultaneously, significantly speeding up the training and inference time of deep learning models [18].

ASICs

While GPUs have general-purpose computation capabilities, ASICs are designed for a specific task. In the context of deep learning, Google's TPUs are ASICs designed specifically for accelerating machine learning workloads. They are tailored to machine learning computations, designed to accelerate matrix operations, which are at the heart of neural network computations. This allows TPUs to provide significant improvements in terms of speed and energy efficiency compared to general-purpose hardware [11].

In conclusion, hardware acceleration is a key method for improving the efficiency of deep learning models. By leveraging GPUs and ASICs, it is possible to significantly reduce the time and resources required to train and deploy these models, enabling their use in a wide range of applications.

3.4 Multi-task Learning

In this direction, researchers attempt to improve efficiency by training a model to complete multiple related tasks (e.g., depth estimation and semantic segmentation). The central idea of this method is that these tasks may have shared underlying features, and joint learning can improve the performance and efficiency of the model.

Multi-task learning (MTL) is a paradigm in machine learning where a model is trained on multiple related tasks simultaneously, thereby improving the performance on individual tasks. The central idea behind MTL is that by learning tasks jointly, the model can leverage commonalities and differences across tasks, leading to improved generalization [3].

Ruder provided a comprehensive overview of multi-task learning, discussing its benefits, challenges, and potential applications. He also discussed various MTL architectures, such as hard parameter sharing and soft parameter sharing, which differ in terms of how much the learned parameters are shared across tasks [22].

In the context of deep learning, several techniques have been proposed to improve the effectiveness of multi-task learning. For instance, Zhang and Yang proposed a novel approach called "sluice networks," which can dynamically control the sharing and partitioning of layers for different tasks in a neural network, offering more flexibility than traditional hard or soft sharing methods [30].

Liu et al. proposed a cross-stitch network for multi-task learning, where the network learns to combine the task-specific representations from multiple tasks. This method allows the model to share what is common between tasks while keeping what is unique to each task separate [16].

Moreover, Kendall et al. proposed a multi-task learning approach using uncertainty to weigh losses for various tasks, allowing the model to effectively prioritize tasks and improve overall performance [12].

In conclusion, multi-task learning is a promising technique for developing lightweight models. By training a model on multiple related tasks simultaneously, it not only reduces the computational resources needed but can also lead to improved generalization and performance on individual tasks.

3.5 Models

Lite-Mono [28] discusses a self-supervised monocular depth estimation model with a hybrid CNN and Transformer architecture. The proposed model incorporates Consecutive Dilated Convolutions (CDC) modules to capture enhanced multi-scale local features and a Local-Global Features Interaction (LGFI) module to calculate the Multi-Head Self-Attention (MHSA) and encode global contexts into the features. The article also explores advanced architectures for depth estimation, such as ResNet, channel-wise attention modules, and feature modulation modules. The proposed model, called Lite-Mono, achieves good results in terms of model complexity, inference speed, and accuracy on the KITTI dataset. The generalization ability of the model is also validated on the Make3D dataset. Ablation studies are conducted to evaluate the importance of different design choices in the architecture.

[15] designs a lightweight monocular depth estimation on edge devices, where it develops a two-stage channel pruning method to, respectively, prune the encoder and decoder based on their characteristics. Extensive experiments show that their strategies are effective on different edge GPU devices, when input resolutions differ in outdoor or indoor scenes.

[14] introduces a lightweight, unsupervised neural network for monocular depth estimation using video sequences. Addressing the challenges of limited labeled data and resource-intensive models, the proposed network efficiently integrates features with fewer parameters. Tested on the KITTI dataset, it outperforms state-of-the-art models in speed and accuracy, particularly on low-end devices like the Raspberry Pi 3. This work paves the way for real-time depth prediction on cost-effective, embedded devices.

In the work by [23], a lightweight encoder-decoder model for monocular depth estimation is introduced, optimized for embedded systems. The distinct feature of this model is the Guided Upsampling Block (GUB), inspired by guided image filtering, which enhances depth map reconstruction. Employing multiple GUBs, the model surpasses competitors in accuracy on the NYU Depth V2 and KITTI datasets and achieves impressive frame rates on NVIDIA platforms.

In [23], the authors introduce a novel method for enhancing monocular depth estimation using 3D points as depth guidance. Distinct from traditional depth completion techniques, their approach excels with sparse, uneven point clouds, making it versatile to the 3D point source. Central to their success is a new multi-scale 3D point fusion network that's streamlined yet effective. Demonstrated on two depth estimation challenges, using structure-from-motion and LiDAR derived 3D points, the network competes with leading depth completion methods in accuracy, especially with minimal points, and stands out in compactness. It also surpasses several modern deep learning multi-view stereo and structure-from-motion techniques in both accuracy and efficiency.

Yet, existing depth estimation techniques, rooted in intricate neural networks, lack real-time efficiency on embedded platforms. Addressing this, the authors in [26] introduce an optimized encoder-decoder archi-

tecture, named FastDepth, enhanced with network pruning to curtail computational demands. Their approach, with a particular emphasis on a low-latency decoder, offers speeds vastly superior to existing methods while retaining comparable accuracy. Specifically, FastDepth boasts 178 fps on NVIDIA Jetson TX2 GPU and 27 fps on its CPU, consuming under 10 W. On the NYU Depth v2 dataset, it closely aligns with top-tier accuracy. The authors assert that their work represents the fastest, most efficient monocular depth estimation on an embedded system suitable for micro aerial vehicles.

4 Results

The models are evaluated on several metrics such as Absolute Relative (Abs Rel), Squared Relative (Sq Rel), Root Mean Square Error (RMSE), RMSE (log), and different scales of delta (δ). Lower values are better for error metrics (Abs Rel, Sq Rel, RMSE, RMSE log), while higher values are better for accuracy metrics ($\delta < 1.25$, $\delta < 1.252$, $\delta < 1.253$). The model size is also provided, with a smaller size being better.

The field of depth estimation has seen significant advancements over the years. As demonstrated by the table, methods have evolved, with newer models achieving lower error rates (Abs Rel, Sq Rel, RMSE, RMSE log) and higher accuracy threshold ($\delta < 1.25$, $\delta < 1.252$, $\delta < 1.253$). The evolution of these methods is also marked by a reduction in the number of parameters (Params), indicative of the growing efficiency and optimization in model design. The most recent model, Lite-Mono [29], from 2023, showcases a combination of impressive performance metrics and a relatively small model size.

Therefore, the field of depth estimation is a rapidly advancing area of research, with improvements seen year after year. The trend towards smaller, more efficient models that maintain, or even increase, performance is clear. This progress, driven by novel methods and techniques, is enabling the deployment of depth estimation models in increasingly resource-constrained environments. Future work will likely continue this trend, seeking to further optimize the balance between model size and performance, while also exploring new applications for these powerful models.

5 Conclusion

The field of monocular depth estimation (MDE) has seen a significant shift towards the development of lightweight models. This trend, as evidenced by the data presented, is driven by the necessity to deploy such models in real-world scenarios with resource constraints, including mobile and embedded applications.

Lightweight models, like Lite-Mono [29] and its variants, have demonstrated that it is indeed possible to achieve impressive performance with a relatively smaller number of parameters. These models embody the principle of efficiency, delivering high-quality depth maps while utilizing fewer computational resources.

In conclusion, the future of MDE seems set to be defined by the development of lightweight models that do not compromise on performance. This trend not only brings forth the potential of greater accessibility and inclusivity, extending the benefits of MDE to lower-end devices and platforms, but also paves the way for the next generation of applications powered by efficient and powerful depth perception capabilities. Future research in this direction is undoubtedly full of potential, and we eagerly anticipate the innovations it will bring.

6 Future Work

In light of the current state of research and the challenges in the field of lightweight monocular depth estimation (MDE), two key areas merit further exploration:

6.1 Domain Adaptation and Robustness in Adverse Conditions

While significant progress has been made in the creation of lightweight MDE models, their performance often degrades in diverse and unseen environments,

Method	Year	Abs Rel	Sq Rel	RMSE	logRMSE	$\delta < 1.25$	$\delta < 1.25^2$	$\delta < 1.25^3$	Params
GeoNet [27]	2018	0.149	1.060	5.567	0.226	0.796	0.935	0.975	31.6M
DDVO [25]	2018	0.151	1.257	5.583	0.228	0.810	0.936	0.974	28.1M
Monodepth2-Res18 [1]	2019	0.115	0.903	4.863	0.193	0.877	0.959	0.981	14.3M
Monodepth2-Res50 [1]	2019	0.110	0.831	4.642	0.187	0.883	0.962	0.982	32.5M
SGDepth [13]	2020	0.113	0.835	4.693	0.191	0.879	0.961	0.981	16.3M
Johnston et al. [10]	2020	0.111	0.941	4.817	0.189	0.885	0.961	0.981	14.3M
CADepth-Res18 [2]	2021	0.110	0.812	4.686	0.187	0.882	0.962	0.983	18.8M
HR-Depth [9]	2021	0.109	0.792	4.632	0.185	0.884	0.962	0.983	14.7M
R-MSFM3 [19]	2021	0.114	0.815	4.712	0.193	0.876	0.959	0.981	3.5M
R-MSFM6 [20]	2021	0.112	0.806	4.704	0.191	0.878	0.960	0.981	3.8M
MonoFormer [17]	2022	0.108	0.806	4.594	0.184	0.884	0.963	0.983	23.9M
Lite-Mono-tiny [29]	2023	0.110	0.837	4.710	0.187	0.880	0.960	0.982	2.2M
Lite-Mono-small [29]	2023	0.110	0.802	4.671	0.186	0.879	0.961	0.982	2.5M
Lite-Mono [29]	2023	0.107	0.765	4.561	0.183	0.886	0.963	0.983	3.1M

Table 1: Comparison of different depth estimation models.

particularly in low-light or adverse conditions. Future research could focus on developing models that exhibit better domain adaptation capabilities to handle these challenges.

Techniques such as transfer learning, unsupervised or semi-supervised learning could be employed to adapt pre-trained models to new domains. Additionally, training models with datasets that encapsulate a variety of conditions (e.g., different lighting, weather, and scene complexities) could help build models that are more robust and capable of handling real-world diversity.

6.2 Uncertainty Estimation and Model Explainability

Current lightweight MDE models mostly focus on providing accurate depth estimates, often overlooking the uncertainty associated with these predictions. Incorporating a measure of uncertainty in the model's outputs can provide valuable context for the predicted depth maps, improving the reliability and interpretability of the model.

On a related note, as MDE models continue to evolve, understanding their decision-making process becomes increasingly critical, especially in safety-critical applications. Future work could focus on making these models more transparent and explainable. Techniques such as feature visualization, saliency mapping, or layer-wise relevance propagation could be used to interpret the model's predictions.

References

[1] Anonymous. Digging into self-supervised monocular depth estimation. *ICLR*, 2019. 9

[2] Firstname Cadepth and etc. Title of the paper. *Journal Name*, xx(yy):pp–pp, 2021. 9

[3] Rich Caruana. Multitask learning. *Machine learning*, 28(1):41–75, 1997. 6

[4] Andreas Geiger, Philip Lenz, Christoph Stiller, and Raquel Urtasun. Vision meets robotics: The kitti dataset. *The International Journal of Robotics Research*, 32(11):1231–1237, 2013. 4

[5] Song Han, Jeff Pool, John Tran, and William Dally. Learning both weights and connections for efficient neural network. *arXiv preprint arXiv:1506.02626*, 2015. 5

[6] Yihui He, Xiangyu Zhang, and Jian Sun. Channel pruning for accelerating very deep neural networks. *arXiv preprint arXiv:1707.06168*, 2017. 5

[7] Geoffrey Hinton, Oriol Vinyals, and Jeff Dean. Distilling the knowledge in a neural network. *arXiv preprint arXiv:1503.02531*, 2015. 5

[8] Andrew G Howard, Menglong Zhu, Bo Chen, Dmitry Kalenichenko, Weijun Wang, Tobias Weyand, Marco Andreetto, and Hartwig Adam. Mobilenets: Efficient

convolutional neural networks for mobile vision applications. *arXiv preprint arXiv:1704.04861*, 2017. 6

[9] Firstname HR-Depth and etc. Title of the paper. *Journal Name*, xx(yy):pp–pp, 2021. 9

[10] Firstname Johnston and etc. Title of the paper. *Journal Name*, xx(yy):pp–pp, 2020. 9

[11] Norman P Jouppi, Cliff Young, Nishant Patil, David Patterson, Gaurav Agrawal, Raminder Bajwa, Sarah Bates, Suresh Bhatia, Nan Boden, Al Borchers, et al. In-datacenter performance analysis of a tensor processing unit. In *Proceedings of the 44th Annual International Symposium on Computer Architecture*, pages 1–12, 2017. 6

[12] Alex Kendall, Yarin Gal, and Roberto Cipolla. Multi-task learning using uncertainty to weigh losses for scene geometry and semantics. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 7482–7491, 2018. 7

[13] Marvin Klingner et al. Self-supervised monocular depth estimation: Solving the dynamic object problem by semantic guidance. In *European Conference on Computer Vision*, 2020. 9

[14] Jun Liu, Qing Li, Rui Cao, Wenming Tang, and Guoping Qiu. Mininet: An extremely lightweight convolutional neural network for real-time unsupervised monocular depth estimation. *ISPRS Journal of Photogrammetry and Remote Sensing*, 166:255–267, 2020. 7

[15] Siping Liu, Laurence Tianruo Yang, Xiaohan Tu, Renfa Li, and Cheng Xu. Lightweight monocular depth estimation on edge devices. *IEEE Internet of Things Journal*, 9(17):16168–16180, 2022. 7

[16] Zhongxing Liu, Jian Gao, Ming Gong, Jie Zhou, and Chang Gong. Multi-task learning as multi-objective optimization. In *NeurIPS*, 2019. 7

[17] Firstname MonoFormer and etc. Title of the paper. *Journal Name*, xx(yy):pp–pp, 2022. 9

[18] John Nickolls, Ian Buck, Michael Garland, and Kevin Skadron. Scalable parallel programming with cuda. In *Queue*, volume 6, pages 40–53. ACM, 2008. 6

[19] Firstname R-MSFM3 and etc. Title of the paper. *Journal Name*, xx(yy):pp–pp, 2021. 9

[20] Firstname R-MSFM6 and etc. Title of the paper. *Journal Name*, xx(yy):pp–pp, 2021. 9

[21] Adriana Romero, Nicolas Ballas, Samira Ebrahimi Kahou, Antoine Chassang, Carlo Gatta, and Yoshua Bengio. Fitnets: Hints for thin deep nets. *arXiv preprint arXiv:1412.6550*, 2014. 5

[22] Sebastian Ruder. An overview of multi-task learning in deep neural networks. *arXiv preprint arXiv:1706.05098*, 2017. 6

[23] Michael Rudolph, Youssef Dawoud, Ronja Güldenring, Lazaros Nalpantidis, and Vasileios Belagiannis. Lightweight monocular depth estimation through guided decoding. In *2022 International Conference on Robotics and Automation (ICRA)*, pages 2344–2350. IEEE, 2022. 7

[24] Mingxing Tan and Quoc V Le. Efficientnet: Rethinking model scaling for convolutional neural networks. *arXiv preprint arXiv:1905.11946*, 2019. 6

[25] Chaoyang Wang et al. Deep direct visual odometry for autonomous vehicles. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018. 9

[26] Diana Wofk, Fangchang Ma, Tien-Ju Yang, Sertac Karaman, and Vivienne Sze. Fastdepth: Fast monocular depth estimation on embedded systems. In *2019 International Conference on Robotics and Automation (ICRA)*, pages 6101–6108. IEEE, 2019. 7

[27] Zhichao Yin and Jianping Shi. Geonet: Unsupervised learning of dense depth, optical flow and camera pose. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018. 9

[28] Ning Zhang, Francesco Nex, George Vosselman, and Norman Kerle. Lite-mono: A lightweight cnn and transformer architecture for self-supervised monocular depth estimation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 18537–18546, June 2023. 7

[29] Ning Zhang, Francesco Nex, George Vosselman, and Norman Kerle. Lite-mono: A lightweight cnn and transformer architecture for self-supervised monocular depth estimation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 18537–18546, 2023. 8, 9

[30] Weining Zhang and Bo Yang. Deep model based domain adaptation for fault diagnosis. In *IEEE Transactions on Industrial Electronics*, volume 65, pages 2444–2452. IEEE, 2018. 7