A Simple Baseline for Fast and Accurate Depth Estimation on Mobile Devices

Ziyu Zhang, Yicheng Wang, Zilong Huang, Guozhong Luo, Gang Yu, Bin Fu
Tencent GY-Lab
{parkzyzhang, yichengwang, zilonghuang, alexantaluo, skicyyu, brianfu}@tencent.com

Abstract

In this paper, we propose a simple but effective encoder-decoder based network for fast and accurate depth estimation on mobile devices. Unlike other depth estimation methods using heavy context modeling modules, the encoder with a fast downsampling strategy is employed to obtain sufficient receptive field and contexts at a faster rate. To obtain dense prediction, a light decoder is adopted to recover back to the original resolution. Additionally, to improve the representative ability of the light network, we introduce a teacher-student strategy. It relies on a distillation process ensuring that the student (the proposed light network) learns from the teacher. The proposed method achieves a good trade-off between latency and accuracy. We evaluated the proposed algorithm on the MAI 2021 Monocular Depth Estimation Challenge and achieved a score of 129.41, ranked the first place, which wins the second by a large margin (129.41 v.s. 14.51). More specifically, the proposed method achieves a si-RMSE score of 0.28 with 97 ms on the Raspberry Pi 4.

1. Introduction

Depth estimation from 2D images has been studied in computer vision for a long time and is nowadays applied to robotics, autonomous driving cars, scene understanding, and 3D reconstructions. As a consequence, this is very problematic to many advanced real-world applications, such as self-driving cars and robot navigation, which desperately demand real-time online depth estimation on mobile devices. Thus, research along the line to make depth estimation run fast while not sacrificing too much quality is gaining increasing attention.

Estimating accurate depth from a single image is challenging because it is an ill-posed problem as infinitely many 3D scenes can be projected to the same 2D scene. However, recent works based on deep convolutional neural networks show great progress with plausible results. After the first learning-based monocular depth estimation work from Saxena et al. [24] was introduced, considerable improvements [5, 6, 16, 17, 18, 19, 20, 23, 26] have been made along with rapid advances in deep learning. While most of the state-of-the-art works apply models based on deep convolutional neural networks (DCNNs) in a supervised fashion, some works proposed semi- or self-supervised learning methods which do not entirely rely on the ground truth depth data. The convolutional neural networks are generally composed of two parts: an encoder for dense feature extraction and a decoder for predicting the desired depth. In the encoder-decoder schemes, some powerful deep networks such as VGG [25],ResNet [7], DenseNet [10] or ShuffleNet [28], Mobilenet [9] are adopted as encoder and a series of strided convolution and spatial pooling layers lower the spatial resolution of transitional outputs, and several techniques such as skip connections or multi-layer deconvolutional networks are adopted to recover back to the original resolution for effective dense prediction. To obtain sufficient receptive field and contexts for better depth estimation, the atrous spatial pyramid pooling (ASPP) [3], pyramid Pooling module (PPM) [29], global convolution network (GCN) [21] and self-attention module [11] have been introduced.

As reported by [13, 14, 15], there was a fast development of the deep learning field, with numerous novel approaches and models that were achieving a fundamentally new level of performance for many practical tasks. At the same time, mobile devices started to get multi-core processors, as well as powerful GPUs, DSPs and NPUs, well suitable for machine and deep learning tasks [27]. However, it is also challenge deploy these networks into mobile devices due to the huge computation.

In this paper, we introduce a simple but effective encoder-decoder architecture for fast and accurate depth estimation on mobile devices. To obtain the sufficient receptive field and contexts at minimal computational cost, we choose a light encoder with a fast downsampling strategy, which could quickly downsample the resolution of input images from 480×640 to 4×6. To recover the spatial details, the light decoder is introduced, which consists of a few convolutional layers and upsampling layers. To further improve the representative ability of the light network, we

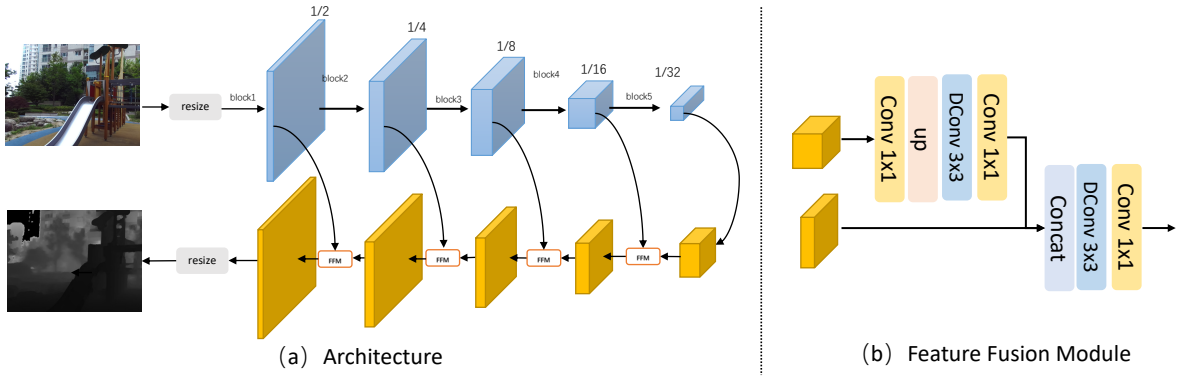

Figure 1. The proposed network architecture.

introduce a teacher-student strategy. It relies on a distillation process ensuring that the student (the proposed light network) learns from the teacher. The proposed method achieves a good trade-off between latency and accuracy. We evaluated the proposed algorithm on the MAI 2021 Monocular Depth Estimation Challenge and achieved a score of 129.41, ranked the first place, which wins the second by a large margin (129.41 v.s. 14.51). More specifically, the proposed method achieves a si-RMSE score of 0.28 with 97 ms on the Raspberry Pi 4.

2. Method

The overall network architecture with the encoder and decoder sub-networks is visualized in Figure 1. We will introduce more details of the architecture and the training process in the next sections.

2.1. Network Architecture

We design a vanilla encoder-decoder style architecture for fast and accurate depth estimation on mobile devices. We use modified mobilenet-v3 [9] as the feature extractor, which has fewer channels than the original version. To reduce the computation, we insert a Resize layer at beginning of mobilenet-v3 to resize the high-resolution input image from 480×640 to 128×160, then the resized images are fed into the convolution layers, which is denoted as *Fast Downsampling Strategy*. The features extracted from the encoder is with the shape $\frac{128}{32} \times \frac{160}{32}$ and equipped with sufficient receptive field and contexts. Thus, we do not need to build a heavy context modeling model on the top of the encoder and it is crucial to achieving high performance. The Fast Downsampling Strategy makes it possible to extract features with sufficient receptive field and rich contexts, at a fast rate. However, the excessive downsampling layers also lose the majority of the spatial details which is also crucial to the depth estimation task. Following the common practices [22, 21], we use the light decoder to gradually recovery the spatial details by fusing the deep features and shallow features. The decoder consists of several decoding stages. At each decoding stage, a Feature Fusion Module is applied to concatenate features from the neighboring blocks in the encoder, which has spatial resolutions of 1/16, 1/8, 1/4, and 1/2. Since we insert a Resize layer at beginning of mobilenet-v3, another Resize layer is appended to the decoder for resizing the depth map into 480×640.

2.2. Distillation Process

Knowledge Distillation, introduced by Hinton et al. [8], refers to the training paradigm in which a student model leverages "soft" labels coming from a strong teacher network. This is the output vector of the teacher's softmax function rather than just the maximum of scores, which gives a "hard" label. Bucila et al. [2] propose an algorithm to train a single neural network by mimicking the output of an ensemble of models. Ba and Caruana [1] adopt the idea of [2] to compress deep networks into shallower but wider ones, where the compressed model mimics the 'logits'. Such training improves the performance of the student model. We choose the ViT-Large [4] as the teacher's backbone and the proposed light network as student's backbone because larger networks tend to have better performance under proper training since they have ample network capacity. Next, we train ViT-Large on the training set first using ImageNet pretrained parameters as initialization. In the training of the distillation process, we fix the trained teacher network and fed images into the light network and the teacher simultaneously. Because there is no softmax function in the network, we use the features before the last activation layer for knowledge distillation. The L2 distance is used as the distillation loss. The student network is trained to optimize the combination of the distillation loss and depth estimation loss [18].

Table 1. Ranking results in the Mobile AI 2021 Monocular Depth Estimation Challenge. All results are evaluated on the online test server. The best results are labeled in red color.

Team	**si-RMSE**↓	RMSE↓	LOG10↓	REL↓	**Runtime(ms)**↓	**Score**↑
Ours	**0.2836**	**3.56**	**0.1121**	**0.2690**	**97**	**129.41**
KX_SMART	0.2602	3.25	0.1043	0.2678	1197	14.51
dujinhua	0.2408	3.00	0.0904	0.2389	1933	11.75
root12321	0.2449	3.02	0.0963	0.2648	2130	10.08
Jacob.Yao	0.2902	3.91	0.1551	0.4700	1275	8.98
helloworld3	0.3128	3.89	0.1242	0.3228	958	8.74
jey	0.2761	9.68	2.3393	0.9951	2531	5.5
zhyl	0.2332	2.72	0.0831	0.2189	6146	4.11
weichi	0.4659	7.56	0.4493	0.5992	582	1.72
shayanj	0.3543	4.16	0.1441	0.3862	3466	1.36
fanhuanhuan	0.2678	5.96	0.3300	0.5152	26494	0.59
faustChok	0.3737	9.08	0.9605	0.8573	9392	0.38

Table 2. The effect of the model size and distillation process.

Model id	Flops	si-RMSE	runtime (ms)	score
1	27.7M	0.4019	54	4.51
2	48.8M	0.3621	89	5.09
3	90.0M	0.3304	99	**6.63**
4	189.0M	0.3005	176	5.64
3+dist	90.0M	0.3141	99	**8.31**

3. Experiments

3.1. Datasets

The total dataset for the MAI challenge [12] contains 7385 pairs of RGB and grayscale depth images. We use 7000 pairs for training and the rest 385 pairs as the local validation set. The original image size is in VGA resolution (480 $\times$ 640), then is resized to 128 $\times$ 160 in the training phase.

3.2. Evaluation Metrics

In this challenge, each submission is validated based on the following two metrics: 1) The quality of the reconstructed results, measured by the invariant standard root mean squared error (si-RMSE). 2) The runtime of the model on the actual target mobile platform, a Raspberry Pi 4 is used in the challenge. The exact scoring formula used in this challenge is provided below:

$$\textbf{Score}(si-RMSE, runtime) = \frac{2^{-20*si-RMSE}}{C*runtime}. \quad (1)$$

where C is a constant normalization factor that does not depend on the submission.

3.3. Implementation details

We train the proposed model on the open-source machine learning library Pytorch. For training, we use Adam optimizer with betas = (0.9, 0.999) and eps=1e-3 , learning is scheduled via polynomial decay from base learning rate 8e-3 with power p = 0.9. The total number of epochs is 500 with batch size = 32 on four NVIDIA V100 GPUs, which takes around 4 hours to train a model. We use mobilenet-v3 as backbone (pretrained on ImageNet), and the decoder part is trained from scratch.

3.4. Model Optimization and TFLite Conversion

The model is trained in PyTorch and converted from PyTorch to tflite. The converting path is PyTorch $\rightarrow$ ONNX $\rightarrow$ Keras $\rightarrow$ tflite. Since we don't use the full size of the original images, the tflite model contains two resize layers (one to resize the input image to 128 $\times$ 160, another to enlarge the depth map to 480 $\times$ 640), which take around 20ms on

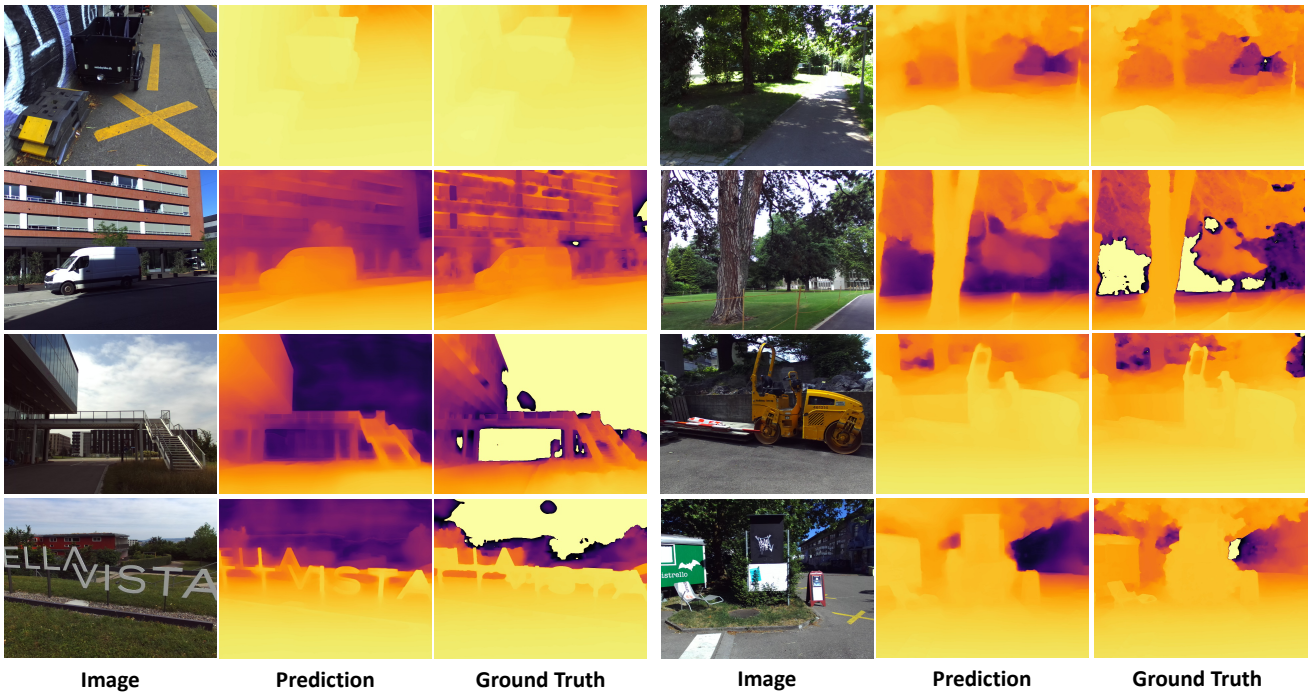

Figure 2. The visualization results of the proposed methods.

raspberry 4. The total runtime of TfLite model on raspberry 4 is shown on Table 1 and Table 2.

3.5. Experimental Results

As shown in Table 1, our proposed method achieves an overall score of 129.41 on the challenge test set and ranks first place. The proposed method achieves 0.2836 si-RMSE with 97 ms on the Raspberry Pi 4. The runtime is substantially lower than the other methods and the performance si-RMSE is comparable to the best performance achieved by the team zhyl.

3.6. Ablation Study

To study the effect of the model size and the distillation process, we show some quantitative results in Table 2. The si-RMSE is measured on the val set which is split by ourselves. The runtime is measured on the online Raspberry Pi 4 provided by the organizers. As shown in Table 2, with increasing the channel numbers in the encoder, the computation cost also increases, and si-RMS decreases gradually. When the computation cost is 90.0M Flops, the model #3 achieves the best score which makes a good trade-off on performance and speed.

To further improve the representative ability of the light network, we introduce a teacher-student strategy. It relies on a distillation process ensuring that the student (the proposed light network) learns from the teacher. Here, we choose model #3 as the student. In Table 2, the model #3 with distillation is denoted as "3+dist" and achieves better performance without introducing extra computation.

3.7. Visualization

To get an intuitive understanding of the proposed, we visualize the prediction results of the proposed methods as shown in Figure 2. The visualization results demonstrate that the proposed method could achieve good depth estimation results. However, the example in the second row, which has a building and car in the image, shows the results predicted by the proposed method are very rough around the edges due to the excessive down-sampling.

4. Conclusion

In this paper, we propose a simple but accurate method for depth estimation, which achieves state-of-the-art results. The well-designed encoder-decoder network makes it possible to extract features with sufficient receptive field and rich contexts. Besides, the distillation process could help improve the performance of the proposed light network without reducing the speed. Finally, the proposed method achieves a good trade-off between latency and accuracy and ranked first place on the MAI 2021 Monocular Depth Estimation Challenge. More specifically, the proposed method achieves a si-RMSE score of 0.28 with 97 ms on the Raspberry Pi 4.

References

[1] Lei Jimmy Ba and Rich Caruana. Do deep nets really need to be deep? *arXiv preprint arXiv:1312.6184*, 2013.

[2] Cristian Buciluǎ, Rich Caruana, and Alexandru Niculescu-Mizil. Model compression. In *Proceedings of the 12th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 535–541, 2006.

[3] Liang-Chieh Chen, George Papandreou, Iasonas Kokkinos, Kevin Murphy, and Alan L Yuille. Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs. *IEEE transactions on pattern analysis and machine intelligence*, 40(4):834–848, 2017.

[4] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929*, 2020.

[5] David Eigen and Rob Fergus. Predicting depth, surface normals and semantic labels with a common multi-scale convolutional architecture. In *Proceedings of the IEEE international conference on computer vision*, pages 2650–2658, 2015.

[6] David Eigen, Christian Puhrsch, and Rob Fergus. Depth map prediction from a single image using a multi-scale deep network. *arXiv preprint arXiv:1406.2283*, 2014.

[7] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016.

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

[9] Andrew Howard, Mark Sandler, Grace Chu, Liang-Chieh Chen, Bo Chen, Mingxing Tan, Weijun Wang, Yukun Zhu, Ruoming Pang, Vijay Vasudevan, et al. Searching for mobilenetv3. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 1314–1324, 2019.

[10] Gao Huang, Zhuang Liu, Laurens Van Der Maaten, and Kilian Q Weinberger. Densely connected convolutional networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4700–4708, 2017.

[11] Zilong Huang, Xinggang Wang, Lichao Huang, Chang Huang, Yunchao Wei, and Wenyu Liu. Ccnet: Criss-cross attention for semantic segmentation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 603–612, 2019.

[12] Andrey Ignatov, Grigory Malivenko, David Plowman, Samarth Shukla, and Radu Timofte. Fast and accurate single-image depth estimation on mobile devices, mobile ai 2021 challenge: Report. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops*, pages 0–0, 2021.

[13] Andrey Ignatov, Radu Timofte, William Chou, Ke Wang, Max Wu, Tim Hartley, and Luc Van Gool. Ai benchmark: Running deep neural networks on android smartphones. In *Proceedings of the European Conference on Computer Vision (ECCV) Workshops*, pages 0–0, 2018.

[14] Andrey Ignatov, Radu Timofte, Andrei Kulik, Seungsoo Yang, Ke Wang, Felix Baum, Max Wu, Lirong Xu, and Luc Van Gool. Ai benchmark: All about deep learning on smartphones in 2019. In *2019 IEEE/CVF International Conference on Computer Vision Workshop (ICCVW)*, pages 3617–3635. IEEE, 2019.

[15] Andrey Ignatov, Radu Timofte, Thang Van Vu, Tung Minh Luu, Trung X Pham, Cao Van Nguyen, Yongwoo Kim, Jae-Seok Choi, Munchurl Kim, Jie Huang, et al. Pirm challenge on perceptual image enhancement on smartphones: Report. In *Proceedings of the European Conference on Computer Vision (ECCV) Workshops*, pages 0–0, 2018.

[16] Seungryong Kim, Kihong Park, Kwanghoon Sohn, and Stephen Lin. Unified depth prediction and intrinsic image decomposition from a single image via joint convolutional neural fields. In *European conference on computer vision*, pages 143–159. Springer, 2016.

[17] Iro Laina, Christian Rupprecht, Vasileios Belagiannis, Federico Tombari, and Nassir Navab. Deeper depth prediction with fully convolutional residual networks. In *2016 Fourth international conference on 3D vision (3DV)*, pages 239–248. IEEE, 2016.

[18] Jin Han Lee, Myung-Kyu Han, Dong Wook Ko, and Il Hong Suh. From big to small: Multi-scale local planar guidance for monocular depth estimation. *arXiv preprint arXiv:1907.10326*, 2019.

[19] Jun Li, Reinhard Klein, and Angela Yao. A two-streamed network for estimating fine-scaled depth maps from single rgb images. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 3372–3380, 2017.

[20] Fayao Liu, Chunhua Shen, Guosheng Lin, and Ian Reid. Learning depth from single monocular images using deep convolutional neural fields. *IEEE transactions on pattern analysis and machine intelligence*, 38(10):2024–2039, 2015.

[21] Chao Peng, Xiangyu Zhang, Gang Yu, Guiming Luo, and Jian Sun. Large kernel matters–improve semantic segmentation by global convolutional network. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4353–4361, 2017.

[22] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. In *International Conference on Medical image computing and computer-assisted intervention*, pages 234–241. Springer, 2015.

[23] Anirban Roy and Sinisa Todorovic. Monocular depth estimation using neural regression forest. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 5506–5514, 2016.

[24] Ashutosh Saxena, Sung H Chung, Andrew Y Ng, et al. Learning depth from single monocular images. In *NIPS*, volume 18, pages 1–8, 2005.

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

[26] Peng Wang, Xiaohui Shen, Zhe Lin, Scott Cohen, Brian Price, and Alan L Yuille. Towards unified depth and semantic prediction from a single image. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2800–2809, 2015.

[27] Kai Zhang, Martin Danelljan, Yawei Li, Radu Timofte, Jie Liu, Jie Tang, Gangshan Wu, Yu Zhu, Xiangyu He, Wenjie Xu, et al. Aim 2020 challenge on efficient super-resolution: Methods and results. *arXiv preprint arXiv:2009.06943*, 2020.

[28] Xiangyu Zhang, Xinyu Zhou, Mengxiao Lin, and Jian Sun. Shufflenet: An extremely efficient convolutional neural network for mobile devices. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 6848–6856, 2018.

[29] Hengshuang Zhao, Jianping Shi, Xiaojuan Qi, Xiaogang Wang, and Jiaya Jia. Pyramid scene parsing network. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2881–2890, 2017.