Enabling monocular depth perception at the very edge

Valentino Peluso, Antonio Cipolletta, Andrea Calimera,
Politecnico di Torino
{valentino.peluso,antonio.cipolletta,andrea.calimera}@polito.it

Matteo Poggi, Fabio Tosi, Filippo Aleotti, Stefano Mattoccia
University of Bologna
{m.poggi,fabio.tosi5,filippo.aleotti2,stefano.mattoccia}@unibo.it

Abstract

Depth estimation is crucial in several computer vision applications, and a recent trend aims at inferring such a cue from a single camera through computationally demanding CNNs — precluding their practical deployment in several application contexts characterized by low-power constraints. Purposely, we develop a tiny network tailored to microcontrollers, processing low-resolution images to obtain a coarse depth map of the observed scene. Our solution enables depth perception with minimal power requirements (a few hundreds of mW), accurately enough to pave the way to several high-level applications at-the-edge.

1. Vision problem

Depth perception is a central and longstanding problem in computer vision, and the recent spread of deep-learning in this area yielded remarkable improvements in this field. Additionally, it also enabled depth perception from a single image with an unprecedented degree of accuracy even through self-supervised training strategies as witnesses by recent works [4, 13, 5, 10, 11]. Indeed, inferring depth from a single image has countless of applications since it does not impose constraint at all the acquisition setup as would occur for other setups such as, for instance, for stereo vision. Nonetheless, despite the efforts carried out recently [8, 12, 9], porting monocular depth estimation on tiny devices with low-power processor cores and few KB of memory, such as microntrollers, is not feasible yet although relevant in practical edge applications and services for the Internet-of-Things (IoT).

2. Low-power solution

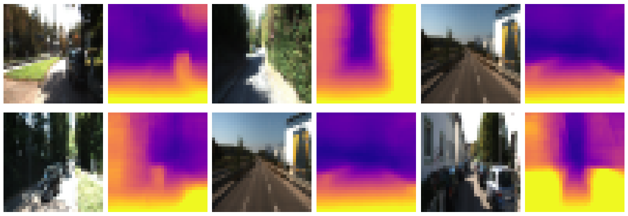

Figure 1. **On-the-edge depth estimation.** Six examples of 32×32 images and inferred depth maps.

Figure 2. **Deployment platform for the proposed solution.** On a microcontroller platform (OpenMV Cam M7), we deploy a compact architecture made of about 100k parameters.

In order to tackle the issues mentioned above, we act according to multiple fronts to enable a sufficiently accurate monocular depth perception system compatible with the constrained computing architectures found at the very edge. By processing low-resolution images (*e.g.*, 48×48), we can infer depth maps as shown in Figure 1 on a sub-W power envelope with the accuracy reported in Table 1.

First and foremost, we propose a new shallow deep network consisting of a lightweight CNN specifically designed for low-resolution images (*e.g.*, 48 × 48 or 32 × 32) according to a pyramidal architecture yielding a depth map of the same size of the input. Another peculiar solution adopted by our method concerns the training procedure, based on a supervisory signal provided by a conventional (*i.e.*, not learning-based) stereo algorithm like Semi Global Matching (SGM [7]). Finally, we adopt for our architecture parallel computation strategies and data quantization specifically suited for ARM microcontrollers. Our network is extremely

			Lower is better				Higher is better		
Method	Resolution	Params	Abs Rel	Sq Rel	RMSE	RMSE log	$\delta < 1.25$	$\delta < 1.25^2$	$\delta < 1.25^3$
PyD-Net [9]	256 × 512	1.9M	0.146	1.291	5.907	0.245	0.801	0.926	0.967
Proposed	48 × 48	0.1M	0.193	2.312	6.952	0.277	0.735	0.890	0.953

Table 1. Quantitative evaluation of PyD-Net [9] and the proposed approach on the test set of KITTI dataset [3] using the split of Eigen et al. [2] with maximum depth set to 80m.

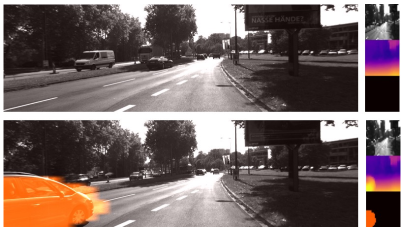

Figure 3. **Simple traffic monitoring application.** At the rightmost position of each frame, we show, from top to bottom, the input 32 × 32 input image, the depth map inferred by our network from it, and the output of a simple detection system working in the depth domain.

compact, with a footprint of about 100kB (*i.e.*, about 100k weights, quantized to 8bit) and trained in a self-supervised manner in order to enable meaningful depth prediction, as reported in Figure 1, at about 2 FPS on the popular ARM M7 architecture (*e.g.*, the OpenMV CAM M7 in Figure 2).

The proposed solution can easily break the 1FPS barrier with less than 0.5W requirements. Respectively, depth estimation carried out on 48×48 and 32×32 images takes about 600 and 300 ms, with a totally memory requirement lower than 512kB. Although some compact solution for depth estimation on embedded systems exist [9, 12], they are far from being competitive in terms of memory footprint, requiring about 2MB only for network weights (*i.e.* 20× larger than our solution).

To the best of our knowledge, our proposal is the first one enabling monocular depth perception on low-power devices such as microcontrollers. Although its accuracy is not comparable to state-of-the-art, we argue that for many high-level applications, a coarse 3D representation of the scene is often enough to manage the faced problem. Whereas state-of-the-art single image depth perception systems [10] might sometimes represent an overkill, our proposal provides meaningful depth maps, as reported in Figure 1, with an accuracy, reported in Table 1, sufficient to a broader range of applications (*e.g.*, people tracking, simple traffic monitoring, etc).

3. Stage of the project

The proposed network has been mapped on different evaluation boards including the OpenMV platform of Figure 2, featuring an STM32F765VI ARM Cortex M7 processor running at 216 MHz with 512KB of RAM, 2 MB of flash and an OV7725 image sensor.

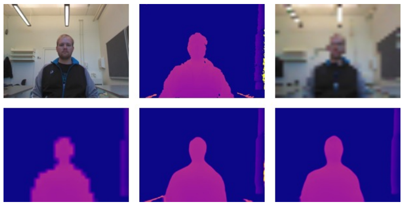

Figure 4. **Privacy-preserving monitoring system.** Top row, from left to right: original VGA image [6], depth map obtained by a Kinect [6], low-resolution image (32 × 32). Bottom row, from left to right: depth map obtained by our network at 32 × 32, the output of network [9] not compatible with the microcontroller, output of a remote up-sampling module fed with the 32 × 32 depth map at the leftmost position.

So far, we have developed two applications relying on the depth maps inferred by our monocular network at the edge. The first one is a simple traffic monitoring system for counting, for instance, cars. The output of such a system is depicted in Figure 3. Having processed a coarse depth map of the monitored environment (top), at any time our system infers depth maps allowing to detect vehicles in the scene, by carrying out the detection in the depth domain. In this case, our focus is on a task with relaxed timing constraints, like traffic congestion monitoring, and not fast decision making needed, like on autonomous driving. However, in case of stringent latency constraints for real-time response, our solution can be easily ported to high-end mobile CPUs (*e.g.*, ARM Cortex-A53) in order to gain about 100× performance at the cost of only 10× power consumption.

The second application concerns a privacy-preserving monitoring system, enabling a simple remote analysis without revealing the identity of the user [1]. Such a system would be useful, for instance, in a hospital to monitor people preserving their privacy by transmitting only the depth maps inferred by the microcontroller. Moreover, the low-resolution depth maps inferred at the edge by the microcontroller, if required, could be upsampled at a higher resolution remotely thought a network hosted in the cloud, where more resources are available. An example of this strategy is depicted in Figure 4. Among advantages, this collaborative edge-cloud strategy allows improving the scalability of the whole infrastructure significantly.

References

[1] E. Chou, M. Tan, C. Zou, M. Guo, A. Haque, A. Milstein, and L. Fei-Fei. Privacy-preserving action recognition for smart hospitals using low-resolution depth images. *arXiv preprint arXiv:1811.09950*, 2018. 2

[2] D. Eigen, C. Puhrsch, and R. Fergus. Depth map prediction from a single image using a multi-scale deep network. In *NIPS*, 2014. 2

[3] A. Geiger, P. Lenz, C. Stiller, and R. Urtasun. Vision meets robotics: The kitti dataset. *IJRR*, 2013. 2

[4] C. Godard, O. Mac Aodha, and G. J. Brostow. Unsupervised monocular depth estimation with left-right consistency. In *CVPR*, 2017. 1

[5] C. Godard, O. Mac Aodha, M. Firman, and G. J. Brostow. Digging into self-supervised monocular depth estimation. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 3828–3838, 2019. 1

[6] R. Hg, P. Jasek, C. Rofidal, K. Nasrollahi, T. B. Moeslund, and G. Tranchet. An rgb-d database using microsoft's kinect for windows for face detection. In *2012 Eighth International Conference on Signal Image Technology and Internet Based Systems*, pages 42–46. IEEE, 2012. 2

[7] H. Hirschmuller. Accurate and efficient stereo processing by semi-global matching and mutual information. In *Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference on*, volume 2, pages 807–814. IEEE, 2005. 1

[8] V. Peluso, A. Cipolletta, A. Calimera, M. Poggi, F. Tosi, and S. Mattoccia. Enabling energy-efficient unsupervised monocular depth estimation on armv7-based platforms. In *Design Automation and Test in Europe (DATE)*, 2019. 1

[9] M. Poggi, F. Aleotti, F. Tosi, and S. Mattoccia. Towards real-time unsupervised monocular depth estimation on cpu. In *IROS*, 2018. 1, 2

[10] F. Tosi, F. Aleotti, M. Poggi, and S. Mattoccia. Learning monocular depth estimation infusing traditional stereo knowledge. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2019. 1, 2

[11] J. Watson, M. Firman, G. J. Brostow, and D. Turmukhambetov. Self-supervised monocular depth hints. In *ICCV*, 2019. 1

[12] Wofk, Diana and Ma, Fangchang and Yang, Tien-Ju and Karaman, Sertac and Sze, Vivienne. FastDepth: Fast Monocular Depth Estimation on Embedded Systems. In *IEEE International Conference on Robotics and Automation (ICRA)*, 2019. 1, 2

[13] T. Zhou, M. Brown, N. Snavely, and D. G. Lowe. Unsupervised learning of depth and ego-motion from video. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, July 2017. 1