Fast Monocular Depth Estimation on an FPGA

2020.5

Tokyo Institute of Technology, Dept. of Information and Communications

Youki Sada, Naoto Soga, Masayuki Shimoda, Akira Jinguji, Shimpei Sato, Hiroki Nakahara

sada@reconf.ict.e.titech.ac.jp, nakahara@ict.e.titech.ac.jp

Outline

1. **Project background**
   - Monocular depth estimation
2. **CNN(Convolutional Neural Network) scheme**
3. **FPGA implementation**
   - Architecture of convolutional circuit
   - Experimental results (accuracy / processing speed)

Monocular Depth Estimation

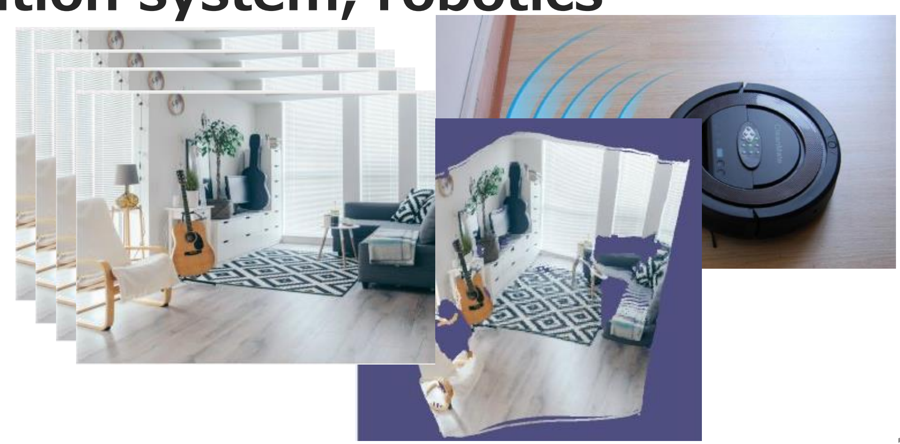
- **Estimate the depth from a single RGB image**

RGB Image

Depth Map

- **Applications: Driving automation system, robotics**

Driving Automation System



SLAM (Simultaneously Localization and Mapping)

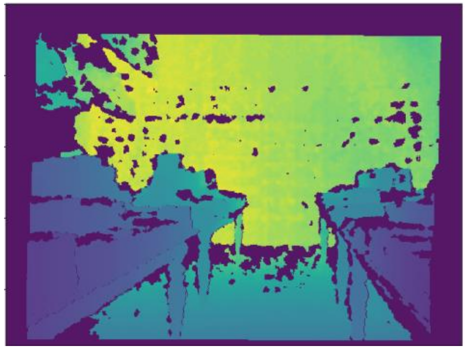
Source: arxiv, Ibraheem et. al, High Quality Monocular Depth Estimation via Transfer Learning

Monocular Depth Estimation (Contd.)

- **Depth sensors**
  - LiDAR
  - Stereo camera

→ Expensive, lack of depth data

LiDAR
(Velodyne)



Stereo camera
(Microsoft Kinect)

- **CNN scheme**
  - Dense Depth Map
  - <u>Use of a monocular RGB camera is valuable</u>

CNN scheme

→ **Implement monocular depth estimation on an FPGA for light-weight and low-cost**

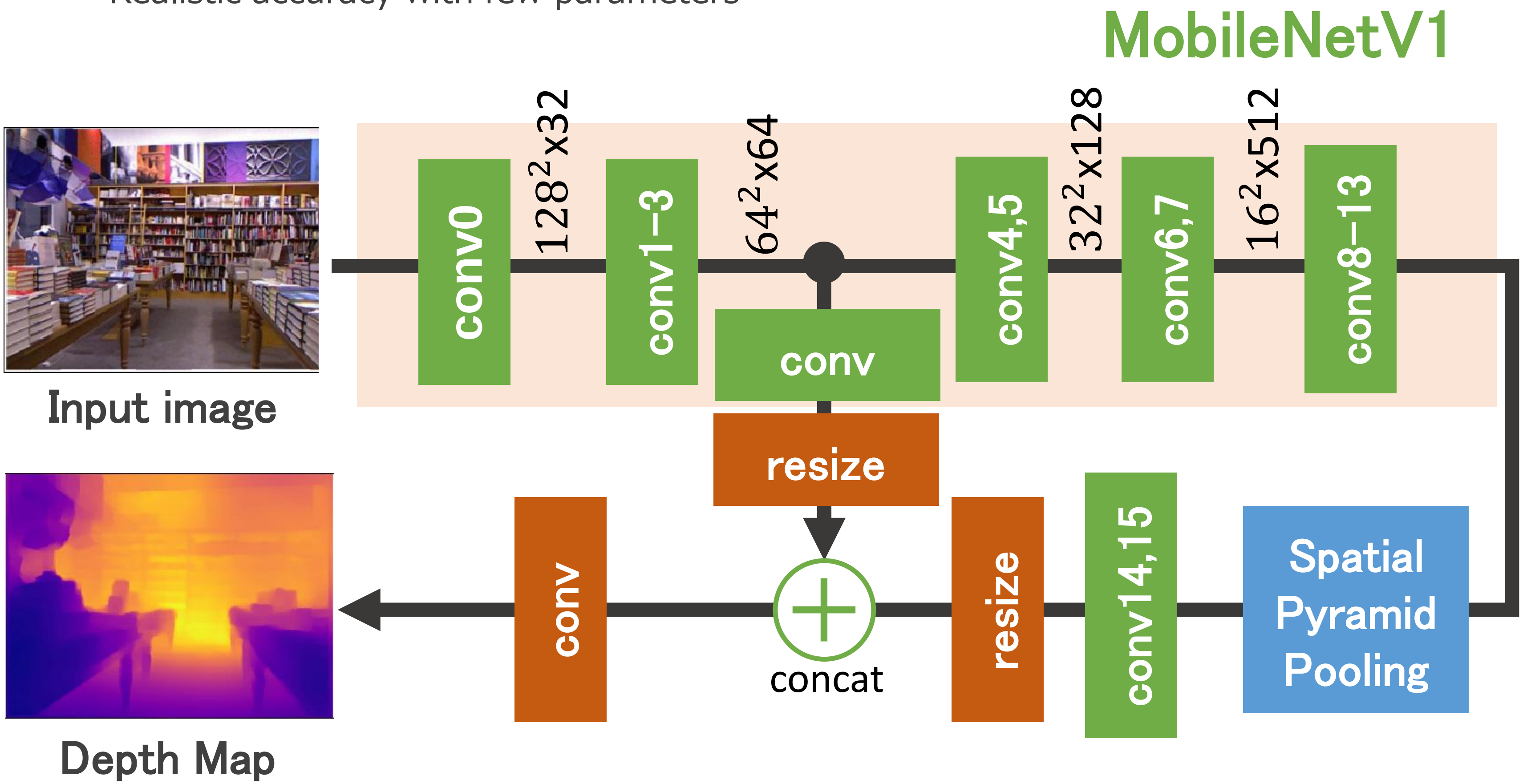
Outline

1. **Project background**
   - Monocular depth estimation
2. **CNN(Convolutional Neural Network) scheme**
3. **FPGA implementation**
   - Architecture of convolutional circuit
   - Experimental results (accuracy / processing speed)

CNN model

- **Convolutional layer only**
  - Depth-CNN (our CNN model)
    - MobileNetV1 base, A skip connection,
      Atrous spatial pyramid pooling (used by SOTA models)

  → Realistic accuracy with few parameters



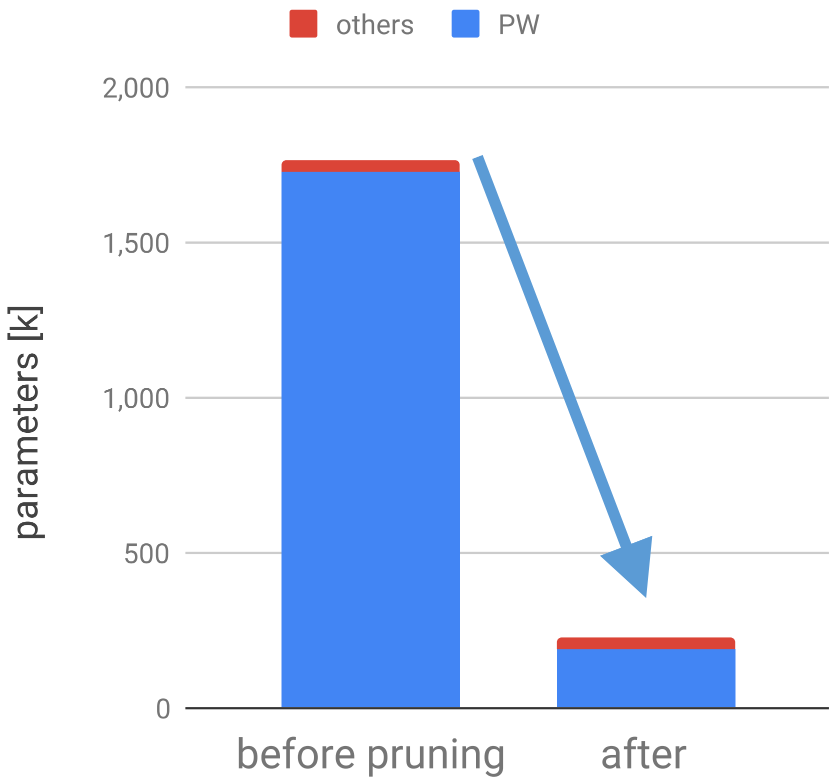
E. Shelhamer, et al., “Fully convolutional networks for semantic segmentation”,
IEEE Transactions on Pattern Analysis and Machine Intelligence, 39(4):640–651, April 2017.

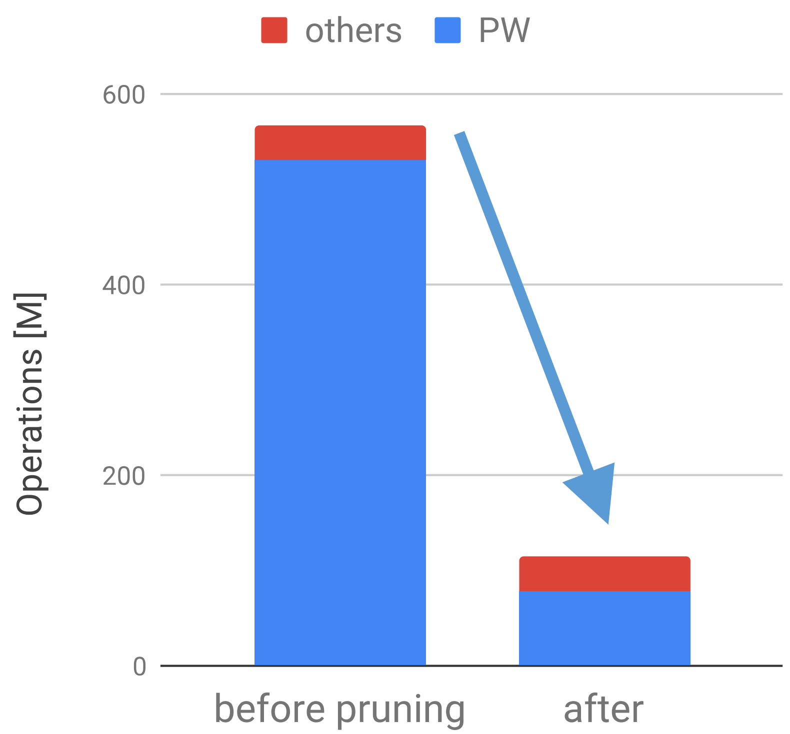
Optimization: Weight Pruning & Quantization

- **Parameters and OPs on PwConv is account for >90%**

  → Apply weight pruning on PwConv (Hardware oriented filter-wise pruning)

- **Weight: 8-bit, Activation: 6-bit Quantization for low-cost FPGA**





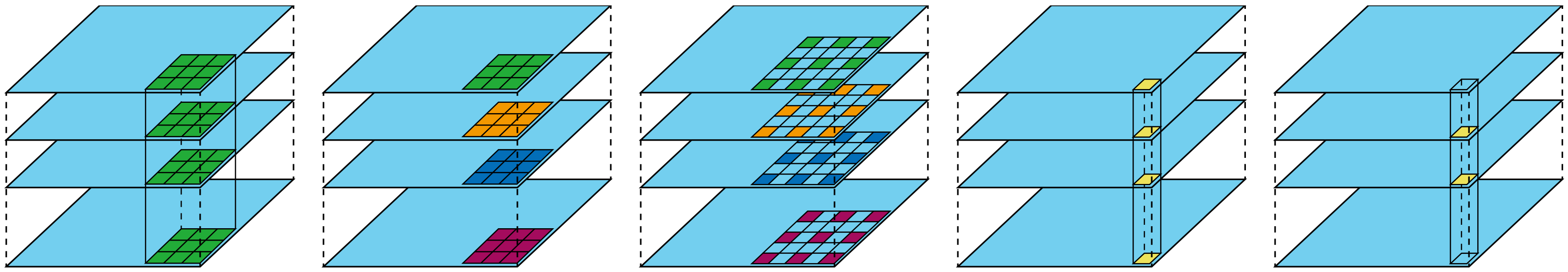
PW; PwConvs, others; Conventional Convs, DwConvs, and Atrous DwConvs

Outline

1. **Project background**
   - Monocular depth estimation
2. **CNN(Convolutional Neural Network) scheme**
3. **FPGA implementation**
   - Architecture of convolutional circuit
   - Experimental results (accuracy / processing speed)

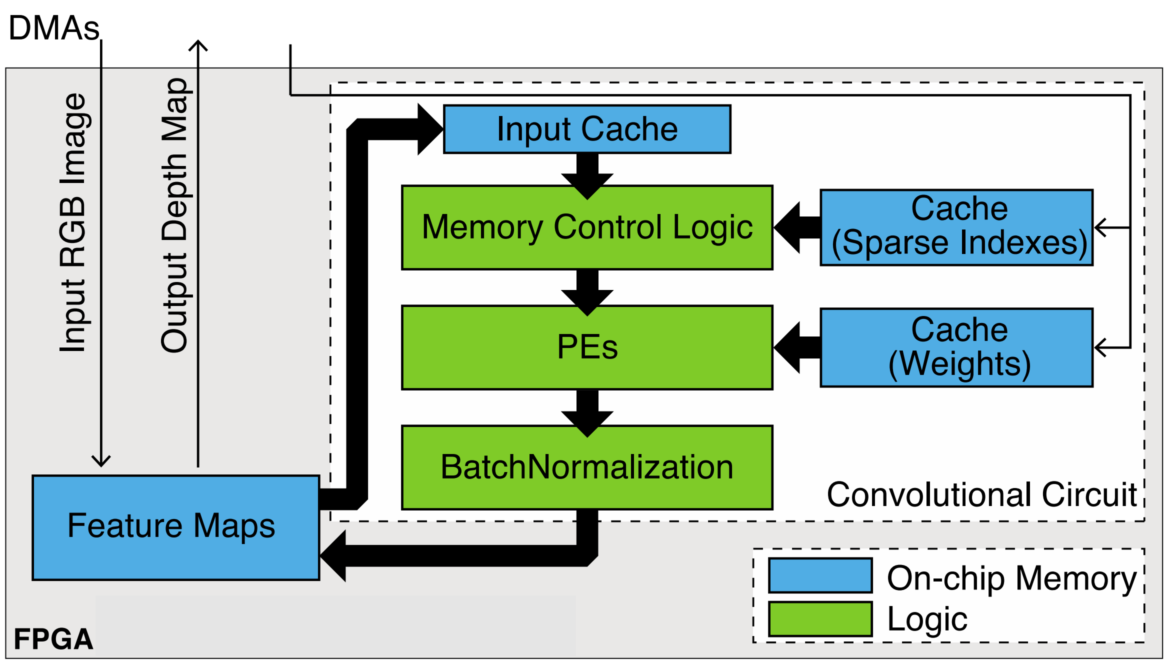
Overall Architecture

- **Single computational engine scheme**
  - general convolution, Depthwise, Atrous Depthwise, Pointwise, Pruned Pointwise



(a) General Convolution (b) Depthwise Convolution (c) Atrous Depthwise Convolution (d) Dense Pointwise Convolution (e) Pruned Pointwise Convolution

- **All calculation results (Feature maps) are stored on on-chip memory**

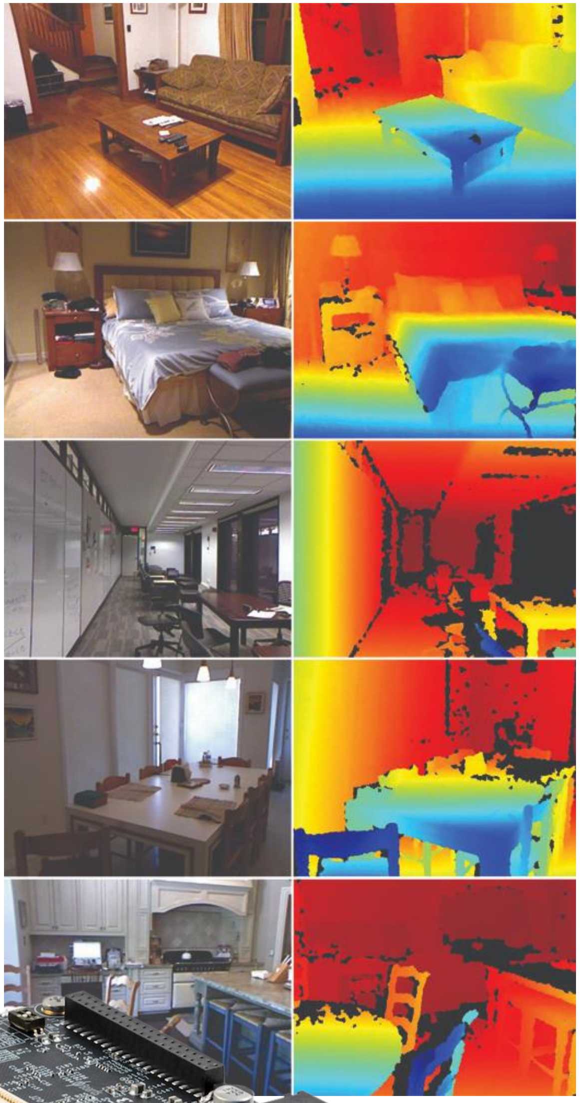


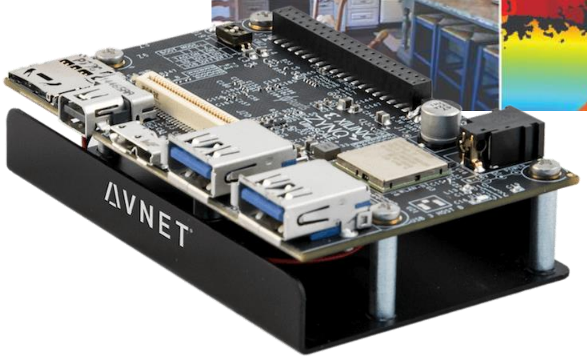
Outline

1. **Project background**
   - Monocular depth estimation
2. **CNN(Convolutional Neural Network) scheme**
3. **FPGA implementation**
   - Architecture of convolutional circuit
   - <u>Experimental results (accuracy / processing speed)</u>

Experiment

- **Dataset: NYU-Depth v2**
  - Generated by Microsoft Kinect
  - Indoor scene images (RGB + Depth Map)
  - 50,688 train images，654 test images
- **FPGA system: Avnet Ultra96**
  - Xilinx Ultrascale+ MPSoC ZU3EG
  - PYNQ Framework and VIVADO HLS
  - Ubuntu
- GPU system: NVIDIA Jetson TX2
  - Tensor RT
  - Ubuntu





Experimental result

Accuracy comparison with conventional CNNs

	Resol.	weights	OP[G]	Accuracy
Eigen[1] (VGG)	228x304	dense	23.4	76.9%
Laina[2] (ResNet50)	228x304	dense	22.9	78.9%
Ours	256x256	dense	0.6	**77.6%**
Ours (wt. 8, act. 6, zero wt. 87%)	256x256	sparse	**0.1**	76.2%

X6 fewer OPs

Comparison for proposed compression

Image size	Precision	Weights	Size[Mb]	OP[M]	Accuracy
256x256	FP32	dense	56.4	0.6	77.6%
256x256	FP32	Sparse	7.2	0.1	76.2%
256x256	wt. 8, act. 6	Sparse	**1.8**	0.1	76.2%

X31 smaller CNN weight size

Achieved X6 fewer OPs and X31 smaller CNN weight size (only 1.4% accuracy degradation)

[1] D. Eigen, C. Puhrsch, and R. Fergus,
"Depth map prediction from a single image using a multi-scale deep network," in Advances in Neural Information Processing Systems (NIPS), 2014, pp. 2366–2374.
[2] I. Laina, C. Rupprecht, V. Belagiannis, F. Tombari, and N. Navab,
"Deeper depth prediction with fully convolutional residual networks," in *2016 Fourth international conference on 3D vision (3DV)*, pp. 239–248, IEEE, 2016.

Experimental result (Contd.)

Utilization result by ZU3EG FPGA

BRAM	DSP	LUT	FF
380 (90%)	198 (55%)	45,666 (65%)	23,254 (16%)

Comparison with GPU

Platform	Compiler	Clock Freq.	Speed [avg. FPS]	Power [W]	Efficiency [FPS/W]
FPGA (Ultra96)	VIVADO HLS	0.2 GHz	**123.6**	**0.3**	**412**
GPU (Jetson TX2)	TensorRT	1.3 GHz	79.8	6.0	13

Compared to the GPU, the FPGA was **X1.5 faster** and **X31 better** performance per power.
Our proposed monocular depth estimation is suitable for embedded systems

Summary

- **Propose lightweight CNN-based monocular depth estimation, which is suitable for embedded systems**
- **Achieved 6X smaller OPs by hardware-oriented weight pruning**
- **Implemented on Ultra96 FPGA board and compared with mobile GPU (Jetson TX2)**
  - Our proposed FPGA system achieved higher efficiency, better speed, and lower power