Table 6: Comparison with software implementations.

Platform	CPU			CPU+GPU			Cloud-DNN-Local		
Device	E5-2430			E5-2609 + Pascal Titan X			VU118		
Model	AlexNet	VGG-16	ResNet-50	AlexNet	VGG-16	ResNet-50	AlexNet	VGG-16	ResNet-50
Data Type	float32			float16			fixed16	fixed16	fixed16
Clock(MHz)	1.9GHz			1GHz			214MHz		
Latency/Image (ms)	242.562	794.238	557.5	5.0486	25.7583	13.79	2.32	16.92	8.12
Speedup(×)	1	1	1	48.05	30.83	40.427	104.55	46.94	68.66

Table 7: Comparison with other designs.

Design	[26]	[20]	[5]	[27]	[28]	Cloud-DNN-AWS	Cloud-DNN-Local
CNN model	VGG16	AlexNet	VGG16-SVD	VGG-19	VGG-16	VGG-16	
Platform	VX690T	VX485T	XC7Z045	Str. V GSMD5	Arr. 10 GX1150	VU9P	
DSPs(used/total)	2833/3600	2240/2800	780/900	1036/1590	1518/1518	5349/6840	
Clock(MHz)	150	100	150	150	200	125	214
Data type	fixed16	float	fixed16	fixed16	fixed16	fixed16	
Power(Watt)	26	18.61	9.63	~25	-	48.62	49.25
Lat./Img.(ms)	65.13	21.61	224.60	-	42.98	28.96	16.92
Thro.(GOPS)	354	61.62	136.97	364.36	720.15	1068.37	1828.61
Eff.(GOPS/W)	13.62	3.31	14.22	14.57	-	21.97	37.13

comparable or better performance to state-of-the-art solutions on FPGAs as well as better energy efficiency compared to CPU and GPU implementations. This framework enables users to quickly create and deploy DNNs on cloud FPGAs. Thus, we provide an efficient and high-performance/energy efficiency FPGA solution for Caffe frameworks in the cloud so users have an additional choice other than always relying on CPU and GPU.

Our workflow is designed in a modular fashion which allows easy extensions for new layer types. There are some potential extensions of this work, such as supporting a wider range of DNNs. Also extending our current flow to support other frameworks like TensorFlow, MXNet and PyTorch is under exploration. We also plan to extend Cloud-DNN to utilize multiple FPGAs in the future. Our current release could be found at https://github.com/microideax/Open-Dnn.git.

ACKNOWLEDGMENTS

This work is partly supported by the National Research Foundation, Prime Minister's Office, Singapore under its Campus for Research Excellence and Technological Enterprise (CREATE) programme, and Alibaba Group through Alibaba Innovative Research (AIR) programme. It is also partly supported by the IBM-Illinois Center for Cognitive Computing System Research (C3SR) - a research collaboration as part of IBM AI Horizons Network.

REFERENCES

[1] Yann LeCun, Yoshua Bengio, and Geoffrey Hinton. Deep learning. *Nature*, 521(7553):436–444, 2015.

[2] Xiaofan Zhang et al. DNNBuilder: an automated tool for building high-performance DNN hardware accelerators for FPGAs. In *Proc. of ICCAD*, 2018.

[3] Hardik Sharma et al. From high-level deep neural models to FPGAs. In *Proc. of MICRO*, 2016.

[4] Jialiang Zhang et al. Improving the performance of OpenCL-based FPGA accelerator for convolutional neural network. In *Proc. of FPGA*, 2017.

[5] Jiantao Qiu et al. Going deeper with embedded FPGA platform for convolutional neural network. In *Proc. of FPGA*, 2016.

[6] Xiaofan Zhang et al. Machine learning on FPGAs to face the IoT revolution. In *Proc. of ICCAD*, 2017.

[7] Junsong Wang et al. Design flow of accelerating hybrid extremely low bit-width neural network in embedded FPGA. In *Proc. of FPL*, 2018.

[8] Huimin Li et al. A high performance FPGA-based accelerator for large-scale convolutional neural networks. In *Proc. of FPL*, 2016.

[9] Naveen Suda et al. Throughput-optimized OpenCL-based FPGA accelerator for large-scale convolutional neural networks. In *Proc. of FPGA*, 2016.

[10] Su Liu et al. Real-time object tracking system on FPGAs. In *Proc. of SAAHPC*, 2011.

[11] Yufei Ma et al. Optimizing loop operation and dataflow in FPGA acceleration of deep convolutional neural networks. In *Proc. of FPGA*, 2017.

[12] Xiaofan Zhang et al. High-performance video content recognition with long-term recurrent convolutional network for FPGA. In *Proc. of FPL*, 2017.

[13] Song Han et al. ESE: Efficient speech recognition engine with compressed LSTM on FPGA. 2016.

[14] Li Qin et al. Implementing neural machine translation with bi-directional GRU and attention mechanism on FPGAs using HLS. In *Proc. of ASP-DAC*, 2019.

[15] Xinheng Liu et al. High level synthesis of complex applications: An h. 264 video decoder. In *Proc. of FPGA*, 2016.

[16] Kyle Rupnow et al. High level synthesis of stereo matching: Productivity, performance, and software constraints. In *Proc. of FPT*, 2011.

[17] Deming Chen et al. Lopass: A low-power architectural synthesis system for FPGAs with interconnect estimation and optimization. *TVLSI*, 18(4):564–577, April 2010.

[18] Andrew Canis et al. LegUp: high-level synthesis for FPGA-based processor/accelerator systems. In *Proc. of FPGA*, 2011.

[19] Emanuele Del Sozzo et al. On the automation of high level synthesis of convolutional neural networks. In *Proc. of IPDPSW*, 2016.

[20] Chen Zhang et al. Optimizing FPGA-based accelerator design for deep convolutional neural networks. In *FPGA*, 2015.

[21] Yongming Shen et al. Maximizing cnn accelerator efficiency through resource partitioning. In *Proc. of ISCA*, 2017.

[22] Xilinx. Large FPGA methodology guide. 2012.

[23] Yangqing Jia et al. Caffe: Convolutional architecture for fast feature embedding. In *Proc. of ACMMM*, 2014.

[24] Song Han et al. Deep compression: Compressing deep neural networks with pruning, trained quantization and huffman coding. 2015.

[25] Philipp Gysel et al. Hardware-oriented approximation of convolutional neural networks. 2016.

[26] Chen Zhang et al. Caffeine: towards uniformed representation and acceleration for deep convolutional neural networks. In *Proc. of ICCAD*, 2016.

[27] Yijin Guan et al. FP-DNN: An automated framework for mapping deep neural networks onto FPGAs with RTL-HLS hybrid templates. In *Proc. of FCCM*, 2017.

[28] Yufei Ma et al. An automatic RTL compiler for high-throughput FPGA implementation of diverse deep convolutional neural networks. In *Prof. of FPL*, 2017.

[29] Roberto DiCecco et al. Caffeinated FPGAs: FPGA framework for convolutional neural networks. In *Proc. of FPT*, 2016.