



PipeCNN: An OpenCL-Based FPGA Accelerator for Convolution Neural Network

Student : Jianjing An and Diankun Jiang Teacher : Dong Wang Team Num: PR022 Institute of Information Science Beijing Jiaotong University

Email: {wangdong, 16112065, 16125141}@bjtu.edu.cn

Jianjing An

PipeCNN —— Introduction

• PipeCNN

PipeCNN is an OpenCL-based FPGA Accelerator for Large-Scale Convolutional Neural Networks (CNNs). There is a growing trend among the FPGA community to utilize High Level Synthesis (HLS) tools to design and implement customized circuits on FPGAs.

Key Features

- A completed OpenCL kernel sets for CNN forward computations
- A generic design, efficient and scalable in performance and cost
- Optimization Design
 - •8-bit fixed-point Design
 - Mixed window/line-buffer caching scheme

PipeCNN —— Top-Level Architecture

Top-Level Architecture

- CNN running on deeply pipelined kernels using Channel/Pipe in OpenCL
- Use a single hardware kernel to implement both the convolution and FC layers



Fig1. The top-level architecture of PipeCNN.

PipeCNN —— Convolution and FC Layers

Convolution:

$$D_0(\mathbf{f}_o, y, x) = \sum_{f_i=0}^{C_l} \sum_{k_y=0}^{K} \sum_{k_x=0}^{K} W_l(f_0, f_i, k_y, k_x) \cdot D_i(f_i, y+k_y, x+k_x)$$

► Inner-product:

$$D_o(f_0) = \sum_{f_i=0}^{C_l} W_l(f_0, f_i) \cdot D_i(f_i)$$

Unified formula:

$$D_0(\mathbf{f}_o, y, x) = \sum_{f_i=0}^{C_i} \sum_{n=0}^{N} W_i(f_0, f_i, n) \cdot D_i(f_i, n) \quad (\mathbf{N} = \mathbf{K} \times K \text{ or } 1)$$



Fig 2. Transform 3D Conv. into 1D accumulation



Fig 3. OpenCL-Modeled Accumulation Circuit

PipeCNN —— Data Mover Kernels

Improving Throughput and Minimizing BW requirements

- Vectorizing feature map and weight
- Utilizing on-chip cache and reusing data in multiple CUs



Fig 4. Data vectorization and reuse in the NDRange

Current Status

- A kernel set supporting state-of-the-art CNN
 - Convolution/FC/pooling/LRN/BN/Relu/Sigmoid/Softmax
 - AlexNet/VGG/NIN/SqueezNet/GoogleNet/ResNet verified
- Tested on main stream FPGA boards
 - Arria-10 (high-end), Stratix-V, Cyclone-V(low-cost)

Imagenet Classification

Accuracy: Table1
Speed: 110ms on
DE10-NANO platform

Table1 The comparison of AlexNet model classification accuracy

Accuracy	Top-1	Top-5
Full precision(32 bit)	56.8%	79.8%
This work(8 bit)	56.2%	79.5%



Fig 5. Imagenet Classification on Alexnet

PipeCNN —— Demonstration2

Object Recognition via Camera

 File
 Edit
 View
 Terminal
 Tabs
 Help

 Pool
 :
 0.000 ms
 ms
 Lrn
 :
 6.719 ms

 Lrn
 :
 0.000 ms
 s
 Lrn
 :
 1.000 ms

 Layer-8:
 MemRd:
 2.023 ms
 conv
 :
 1.863 ms

 Pool
 :
 0.000 ms
 ms
 MemWr:
 1.738 ms

 Lrn
 :
 0.000 ms
 ms
 MemWr:
 1.738 ms

Total kernel runtime 123.268 ms Batch size = 1, average process time per batch: 123.268 ms

Copyed all batched results from fc_2 buffers. Selected item = 0 from the combined batch results in fc buffers

The inference result is n04350905 suit, suit of clothe (the prob is 95.00)

Loading camera



Fig 6. Object recognition via camera on Alexnet

Face Recognition

Datasets: LFW



PipeCNN —— Demonstration4

Object Detection → Full precision mAP: 56.2 → 8-bit mAP: 54.5



Fig 8. Object Detection based on Faster R-CNN(Alexnet)

PipeCNN —— Results



Fig. 9 Design space exploration for AlexNet model on Stratix-V A7 FPGA board. CU denotes compute units, and VEC_SIZE represents the degree of data parallelism utilized. (a) Logic elements utilization; (b) DSP blocks utilization; (c) Inference time.



Fig .10 Resource utilization of each kernel for AlexNet model

Comparison with software accelerators on mobile CPU/GPU*

Table. 2 Summary of the measured performance and power consumption on different platforms

Platform	Frequency	Inference Time ^b	Effective Power ^c	System Power ^d
ARM Cortex ^a A57/A53 CPU	1.9 Ghz (A57) 1.3 Ghz (A53)	20,767 ms	2.4 W	4.1 W
Mali-T760 GPU	700 Mhz	482 ms	0.52 W	2.3 W
Cyclone A5 SoC-FPGA	800 Mhz (CPU) 140 Mhz (FPGA)	110 ms	0.5 W	2.1 W

- ^a Samsung Galaxy Note 4 (Exynos 5433)
- ^b AlexNet benchmark was used.
- ^c Effective power = total power standby power
- ^d Measured by using external power meter with screen turned off

* Oskouei S S, Golestani H B, Hashemi M. CNNdroid: GPU-Accelerated Execution of Trained Deep Convolutional Neural Networks on Android, ACM Conference on Multimedia 2016.

Comparison with HLS/OpenCL-based designs

Table. 2 Summary of the measured performance and power consumption on different platforms

	FPGA2015	FPGA2016	FPGA2017	Our Work	
Device	Virtex-7 VX485T(28nm)	Stratix-V GXA7(28nm)	Arria-10 AX1150(20nm)	Stratix-V GXA7(28nm)	
FPGA Capacity	485K LUTs 2,800 DSPs	622K LEs 256 DSPs	1,150K LEs 2,800 DSPs	622K LEs 256 DSPs	
Frequency	100MHz	120MHz	303MHz	200MHz	
Precision	Float(32b)	fixed(8b-16b)	float(16b) 4 × Improvement	Fixed(8b)	
Inference Time ^a	21.6 ms ^b	45.7 ms		10.5 ms	
Throughput	61.6 GOPS ^b	31.8 GOPS	1,382 GOPS	133.2 GOPS	
DSP Consumed	2,240	246	1,476	247	
Perf. Density (GOPS/DSP/GHz/W)	0.015	0.042	0.068	0.103	
Power	18.6 W	25.8 W	45 W 1.	5x 26.2 W	

^a alexNet model is used.

^b Convolution operation only.

PipeCNN —— Open-Source

doonny / Pip	eCNN					•	♥ Watch ▼	42 🔇	🖈 Unstar	294	¥ Fork	140
Code 🕕	ssues 15)ា Pu	II requests 0	🔟 Insights								
OpenCL-base	d FPGA Ac	celerato	or for Convo	olutional Neural N	letworks							
pencl fpga-a	ccelerator	hls	hardware	altera-opencl-sdk	fpga	deep-learning	deep-ne	eural-netv	vorks			
P 46 com	mits		12 1 branch								acho-2.0	
0 40 0000			PIDIATIO	1	O releases	5	🎎 3 contr	nbutors		₫s Ap	ache-2.0	
ranch: master •	New pull	request	p I branch		C 0 releases	Create nev	v file Upl	oad files	Find file	مِه Ap	e or down	load -
anch: master ▼ doonny fix bu	New pull	request	gr and host of	ada	© 0 releases	Create nev	v file Upl	oad files	Find file	دامم Clone	e or downl 3a9b7 on 2	load ▼ 22 Apr
doonny fix bu	gs a	request add devie	ce and host or	ode nel/host files, impro	♥ 0 releases	Create new	v file Upl	oad files	Find file	درمه مع	e or down 3a9b7 on 2 2 year 5 month	load ▼ 22 Apr rs ago ns ago
doonny fix bu common data documents	s s s s s s s s s s s s s s s s s s s	request add devia add dem	ce and host co o, update ker D17 paper	ode nel/host files, impro	♥ 0 releases	Create new	v file Up	oad files	Find file	م Ap	e or down Ba9b7 on 2 2 year 5 month 4 month	load ▼ 22 Apr rs ago ns ago ns ago
doonny fix but common data documents	s New pull gs a a f	request add devia add dem add fpt20 ix bugs	ce and host co o, update ker D17 paper	ode nel/host files, impro	♥ 0 releases	Create new	v file Up	oad files	Find file	مله Ap	e or down Ba9b7 on 2 2 year 5 month 4 month 2 month	load ▼ 22 Apr rs ago ns ago ns ago
doonny fix bu common data documents project) LICENSE	gs a f I	d request add devia add dem add fpt20 ix bugs nitial con	ce and host co o, update ker D17 paper mmit	ode nel/host files, impro	♥ 0 releases	Create new	v file Upl	oad files	Find file	مله Ap	e or down 3a9b7 on 2 2 year 5 month 2 month 2 month 2 year	load ▼ 22 Apr rs ago ns ago ns ago rs ago

PipeCNN

About

PipeCNN is an OpenCL-based FPGA Accelerator for Large-Scale Convolutional Neural Networks (CNNs). There is a growing trend among the FPGA community to utilize High Level Synthesis (HLS) tools to design and implement customized circuits on FPGAs. Compared with RTL-based design methodology, the HLS tools provide faster hardware development cycle by automatically synthesizing an algorithm in high-level languages (e.g. C/C++) to RTL/hardware. OpenCL[™] is an open, emergying cross-platform parallel programming language that can be used in both GPU and FPGA developments. The main goal of this project is to provide a generic, yet efficient OpenCL-based design of CNN accelerator on FPGAs. Our design is scalable both in performance and hardware resource, and thus can be deployed on a variety of FPGA platforms.

Fig. 11 open-source PipeCNN github



PipeCNN

https://github.com/doonny/PipeCNN

Thanks ~

Backgrounds

The Advantages of OpenCL

- Cross Vendor/Architecture/Device Support
 - Xilinx, Altera, Intel, AMD, Nvidia, ARM, TI
 - FPGA, CPU, GPU, DSP, Many-Core
- High-Level Programming Language/Interface
 - C, C++, Python, Java
- Fast Design, Verification, Test
 - From months to hours
- Ecosystem
 - clBlas, clFFT, clSPARSE, TensorFlow, Caffe
- Integrated with RTL-based Flow for FPGA
 - Wrap RTL modules as kernel functions

