CoDeNet: Efficient Deployment of Input-Adaptive Object Detection on Embedded FPGAs

Qijing Jenny Huang*, Dequan Wang*, Zhen Dong*, Yizhao Gao [†], Yaohui Cai [‡], Tian Li [‡], Bichen Wu, Kurt Keutzer, John Wawrzynek

University of California, Berkeley
[†]University of Chinese Academy of Science
[‡]Peking University

Berkeley
UNIVERSITY OF CALIFORNIA

Outline

- Motivation
- Deformable Convolution
- Operation Codesign
- Detection System Codesign
- Results

Embedded Computer Vision

Applications

Robots

Drones

Autonomous Vehicles

Security Cameras

Mobile Phones

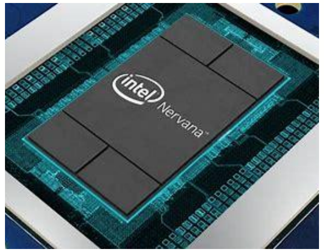
CV Kernels/Tasks

Image Classification

Object Detection

Semantic Segmentation

Embedded Platforms



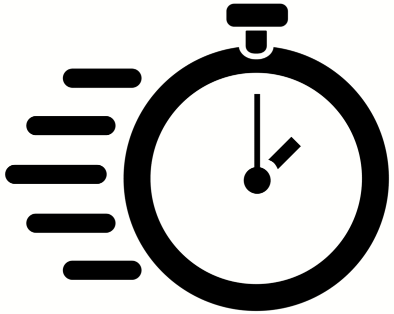
Motivation

- Deployment challenges:
  1. Inefficient Model Designs
  2. Limited compute resources
  3. Real-time requirements
- Object detection:
  - More sensitive to **spatial variance** of objects compared with image classification

Goals

Accuracy

Efficiency



- Codesign **algorithms** and **accelerators** that satisfy embedded system constraints and are pareto-optimal for the accuracy-latency tradeoffs.

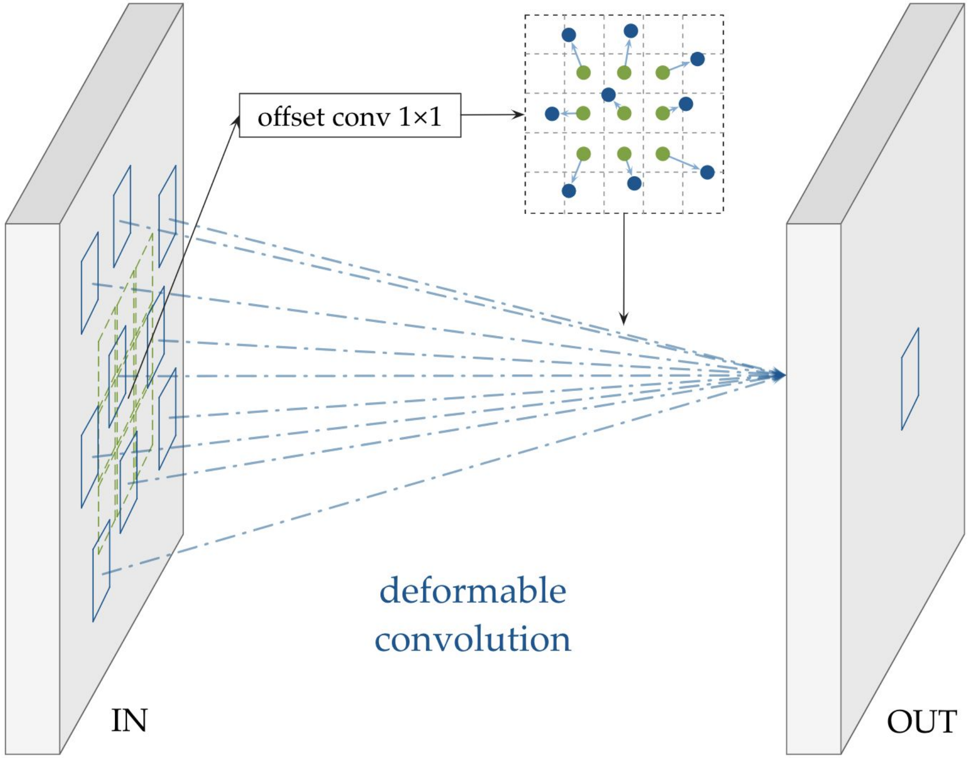
Outline

- Motivation
- Deformable Convolution
- Operation Codesign
- Detection System Codesign
- Results

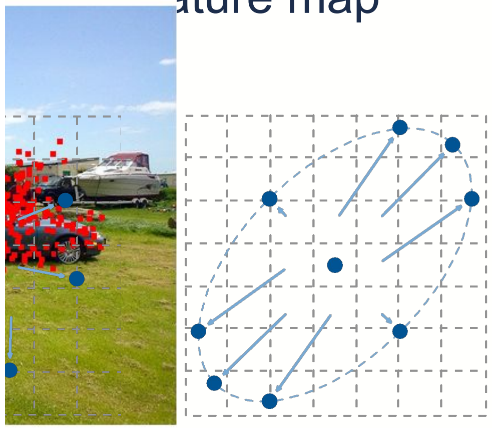
Deformable Convolution

- **Deformable Convolution** is an input-adaptive dynamic operation that samples inputs from variable spatial locations
- Its sampling locat
    - Different input in
    - Different output
- It captures the sp
    - Scales
    - Aspect Ratio
    - Rotation Ang
- Challenges:
    - Increased c
    - Irregular inp
        - Not frien

Sam



1. Generate offsets
2. Sample from input feature map



(in green)

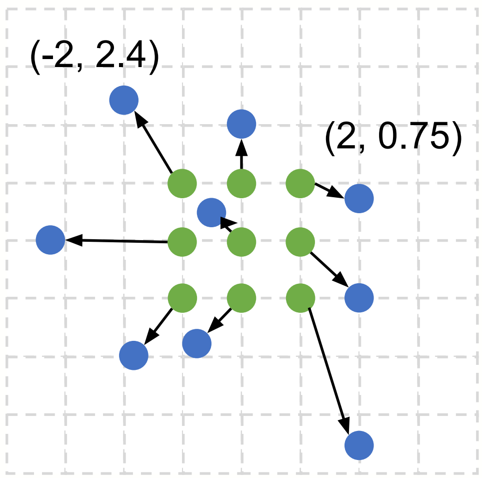
ceptive Fields

Outline

- Motivation
- Deformable Convolution
- Operation Codesign
- Detection System Codesign
- Results

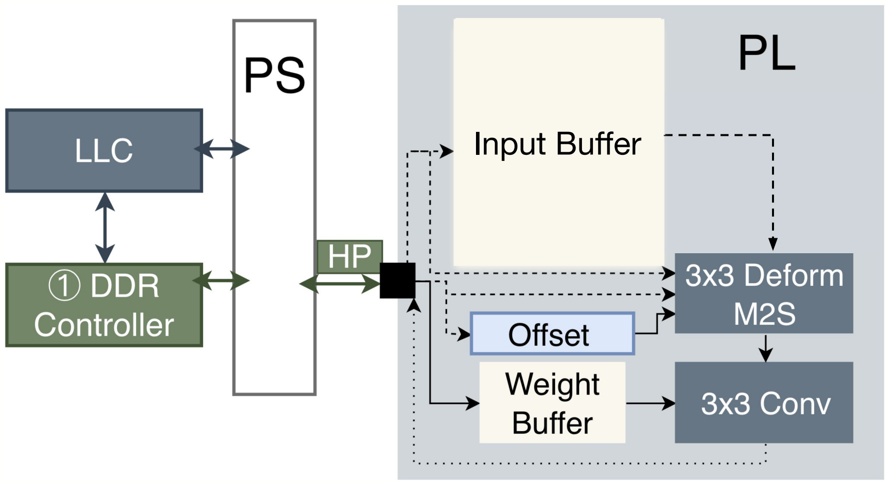
Operation Codesign

Algorithm Modification:



0. Original Deformable

Hardware Optimization:

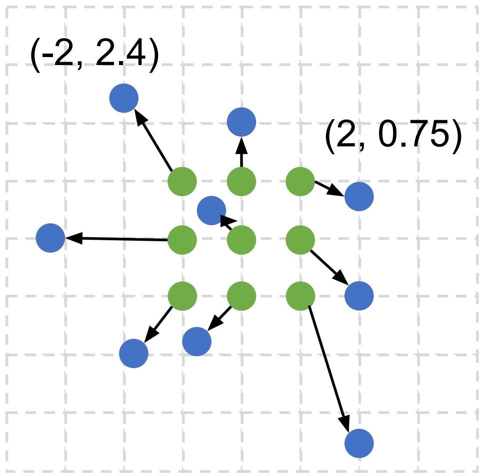


- **Preloads weights to on-chip buffer**
- **Loads input and offsets directly from DRAM**

[1] Accuracy for Object Detection on Pascal VOC

Operation Codesign

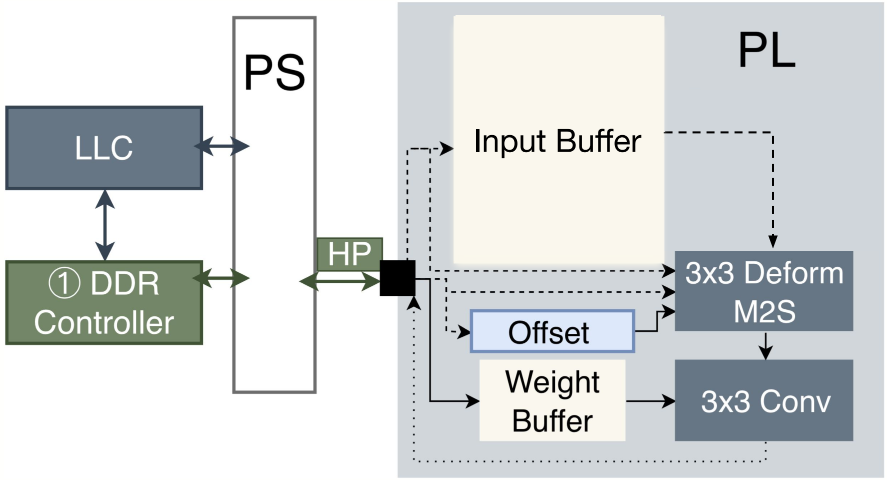
Algorithm Modification:



1. Depthwise Deformable

Accuracy [1](AP): **42.9**

Hardware Optimization:

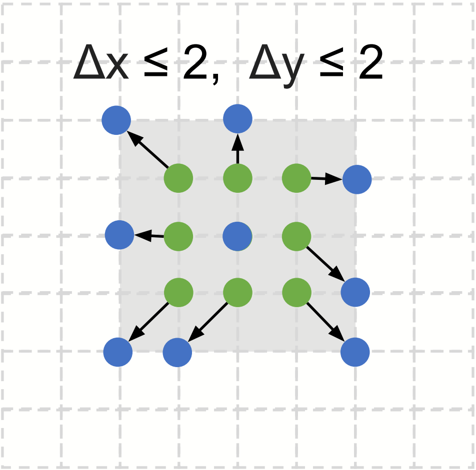


- **Reduce the total MACs**

[1] Accuracy for Object Detection on Pascal VOC

Operation Codesign

Algorithm Modification:

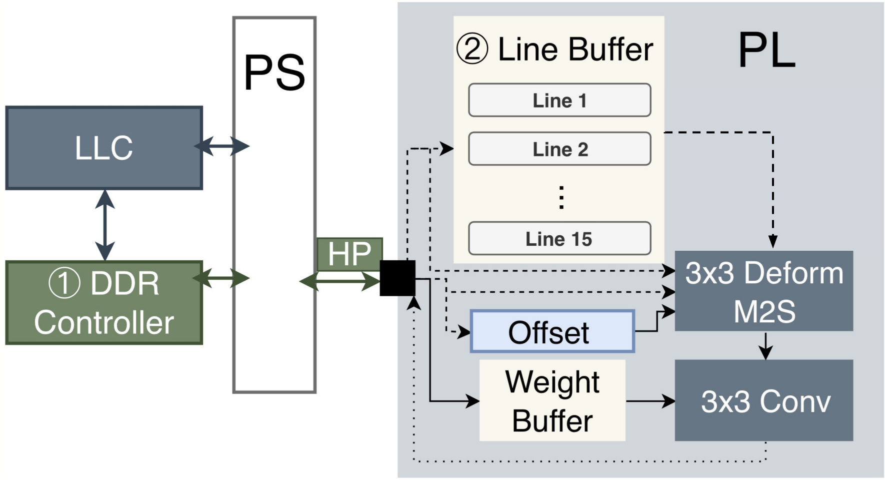


2. Bounded Range

↓ **1.9**

Accuracy [1](AP): **41.0**

Hardware Optimization:

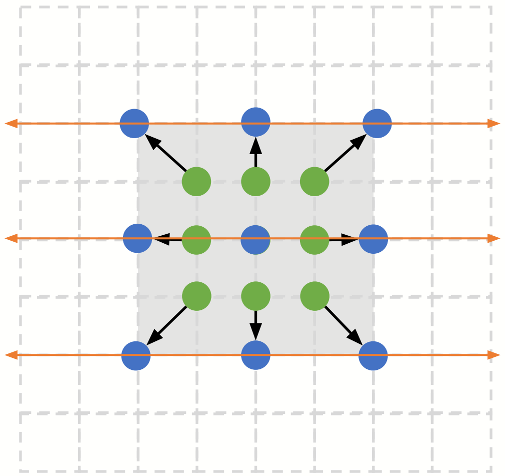


- **Buffers inputs in the on-chip line buffer to allow spatial reuse**

[1] Accuracy for Object Detection on Pascal VOC

Operation Codesign

Algorithm Modification:

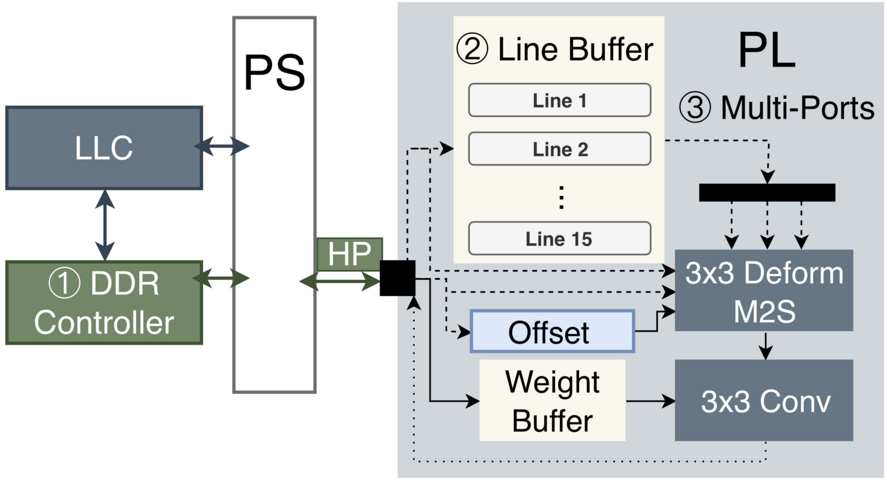


3. Rectangular Shape

↑ **0.1**

Accuracy [1](AP): **41.1**

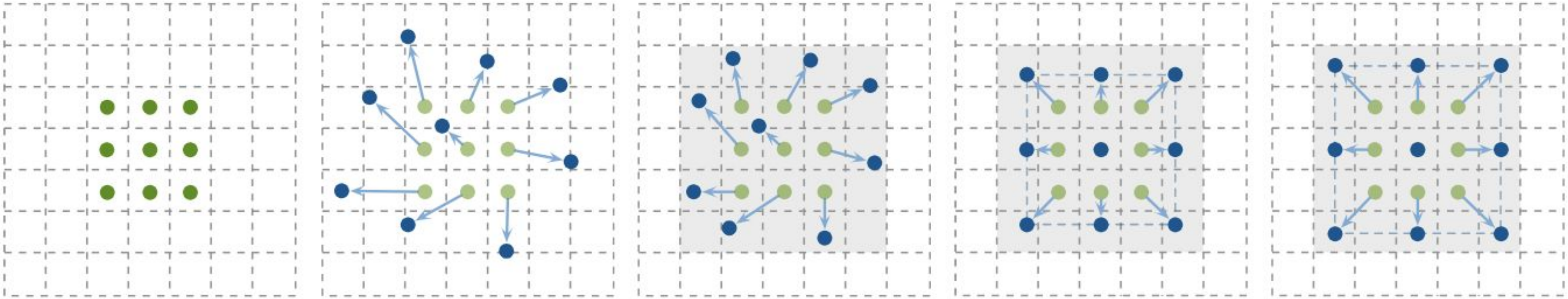
Hardware Optimization:



- **Improves on-chip memory bandwidth**

[1] Accuracy for Object Detection on Pascal VOC

I. Algorithm Modifications



(a) Regular (b) Deformable (c) Bound d) Square. (e) Round

c. **Bounded Range** restricts the range of offsets

d. **Square Shape** limits the geometry to a rectangle shape

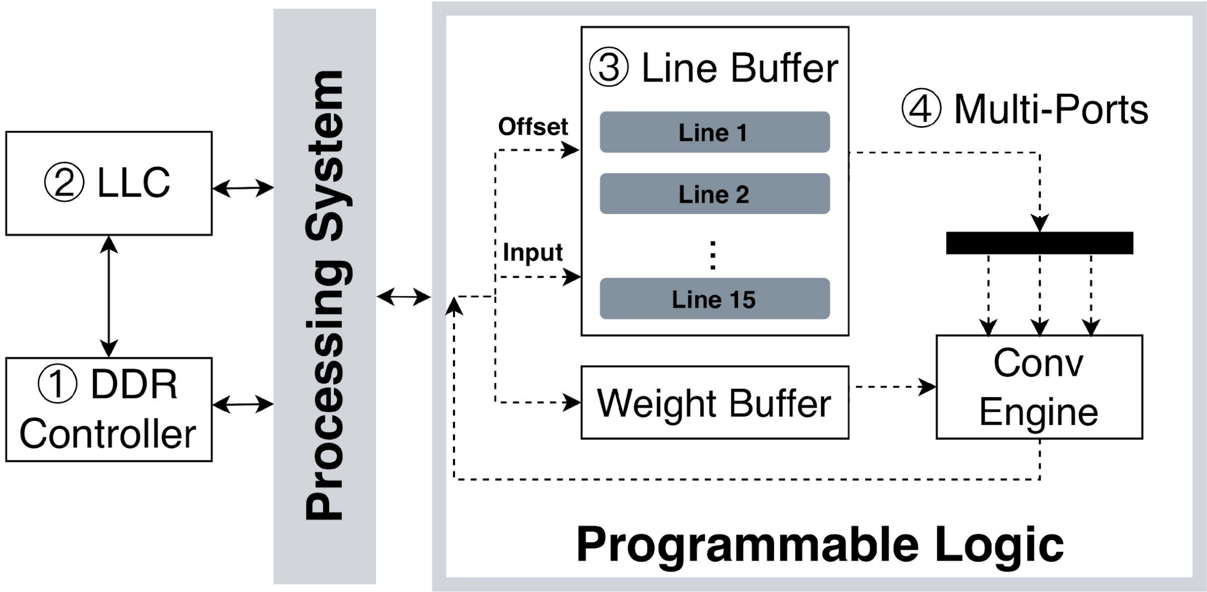
e. **Rounded Offsets** rounds the fractional offsets to integer

f. **Depthwise** replaces full conv with 3x3 dw

Berkeley
UNIVERSITY OF CALIFORNIA

- **More regular access patterns**
- **More efficient operators**

II. Hardware Optimizations



(2) Caching (3) Buffering (4) Parallel Ports

a. **Baseline** loads input features with dynamic offsets from DRAM directly

b. **Caching** adds LLC to leverage temporal and spatial locality

c. **Buffering** uses on-chip BRAM to buffer all inputs from limited range

d. **Parallel Ports** increases on-chip bandwidth with constrained shape

- **Exploit locality**
- **Exploit on-chip bandwidth**

Operation Accuracy

Object Detection Accuracies

Operation	Depthwise	Bound	Square	VOC AP	VOC AP50	VOC AP75	COCO AP	COCO AP50	COCO AP75	COCO APs	COCO APm	COCO APl
3×3				39.2	60.8	41.2	21.4	36.5	21.5	7.3	24.1	33.0
3×3	✓			39.1	60.9	40.9	19.8	34.3	19.7	6.3	22.6	31.5
5×5	✓			40.6	62.4	42.6	21.3	36.4	21.3	6.7	23.7	34.2
7×7	✓			41.9	63.8	43.8	21.7	37.2	21.5	6.9	24.0	35.2
9×9	✓			42.3	64.8	44.3						
deform	✓			42.9	64.4	45.7	23.0	38.4	23.3	6.9	24.4	37.8
deform	✓	✓		41.0	63.0	42.9	21.3	36.4	21.1	7.2	23.6	34.4
deform	✓	✓	✓	41.1	63.1	43.7	21.5	36.8	21.5	6.5	23.7	34.8

5x less compute

< 2 AP change

- 3x3 deformable conv is more efficient than large convolution kernels
- The codesigned deformable conv still achieves good accuracy
  - 1.8 AP difference on Pascal VoC and 1.5 AP difference on COCO

Berkeley
UNIVERSITY OF CALIFORNIA

Operation Performance

Hardware Performance

Operation	Original	Deformable	Bound (buffered)	Square (multi-ported)	Without LLC Latency (ms)	Without LLC GOPs	With LLC Latency (ms)	With LLC GOPs
Full 3×3 Conv	✓				43.1	112.0	41.6	116.2
		✓			59.0	81.8	42.7	113.1
		✓	✓		43.4	111.5	41.8	115.5
		✓	✓	✓	43.4	111.5	41.8	115.6
Depthwise 3×3 Conv	✓				1.9	9.7	2.0	9.6
		✓			20.5	0.9	17.8	1.1
		✓	✓		3.0	6.2	3.4	5.5
		✓	✓	✓	2.1	9.2	2.3	8.2

1.36x

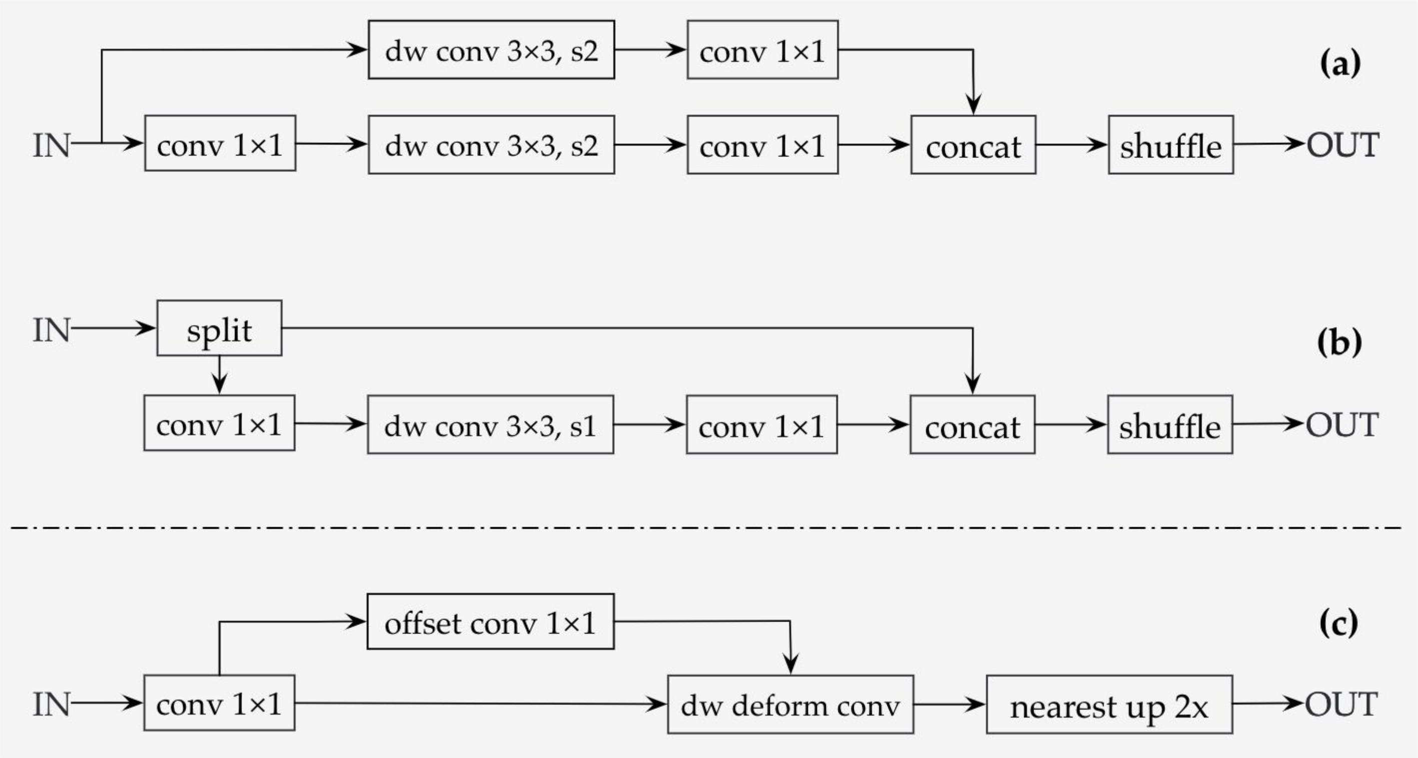
9.76x

1.36× and **9.76×** speedup for full and depthwise deformable conv

Outline

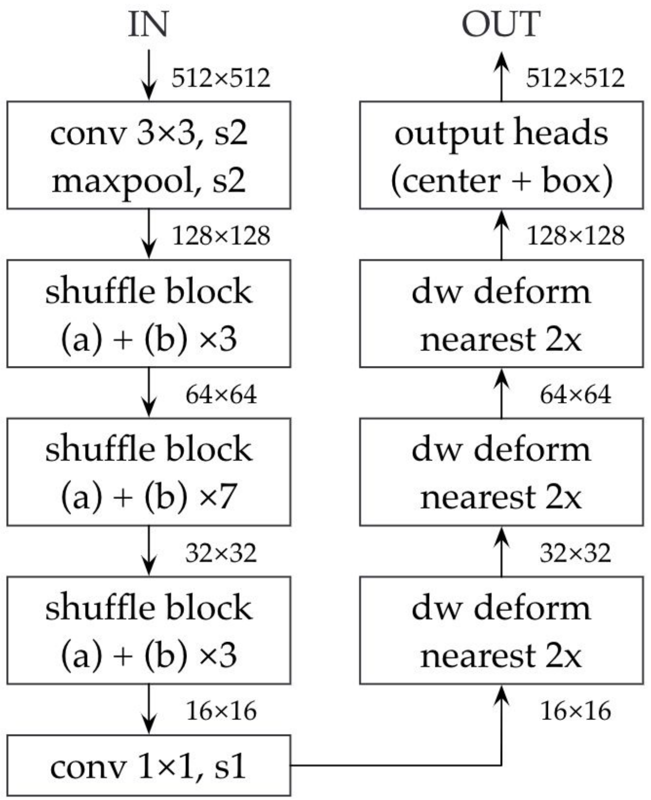
- Motivation
- Deformable Convolution
- Operation Codesign
- Detection System Codesign
- Results

Building Blocks



- **Simple** building blocks to reduce the hardware complexity

ShuffleNetV2 + CenterNet[1]

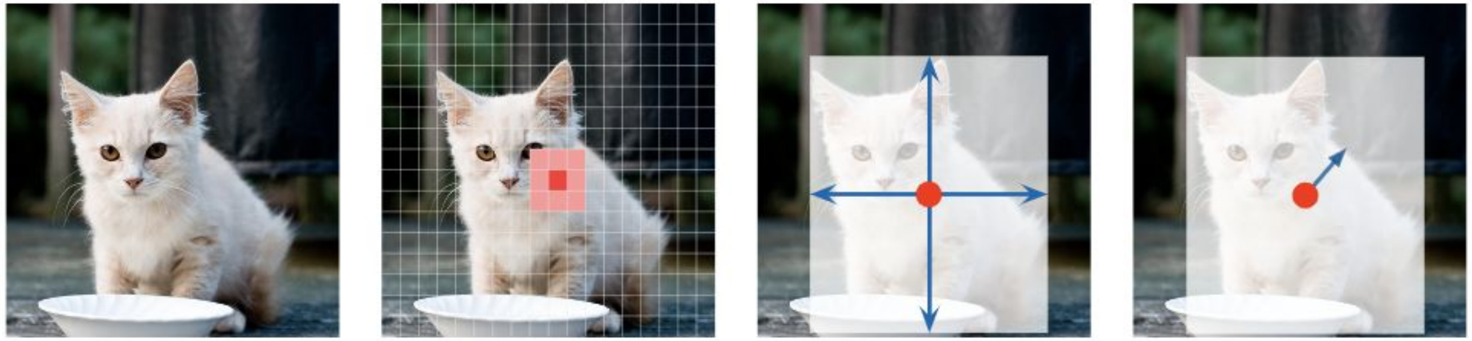


- **Anchor-free** detection system to reduce the postprocessing overhead for Non Maximum Suppression (NMS)

Berkeley
UNIVERSITY OF CALIFORNIA

1 Zhou, Xingyi, Dequan Wang, and Philipp Krähenbühl. "Objects as points." *arXiv preprint arXiv:1904.07850* (2019).

Detection Heads



(a) image (b) center heatmap (c) width & height (d) local shift

1. The center heatmap
2. The object size
3. The local offset

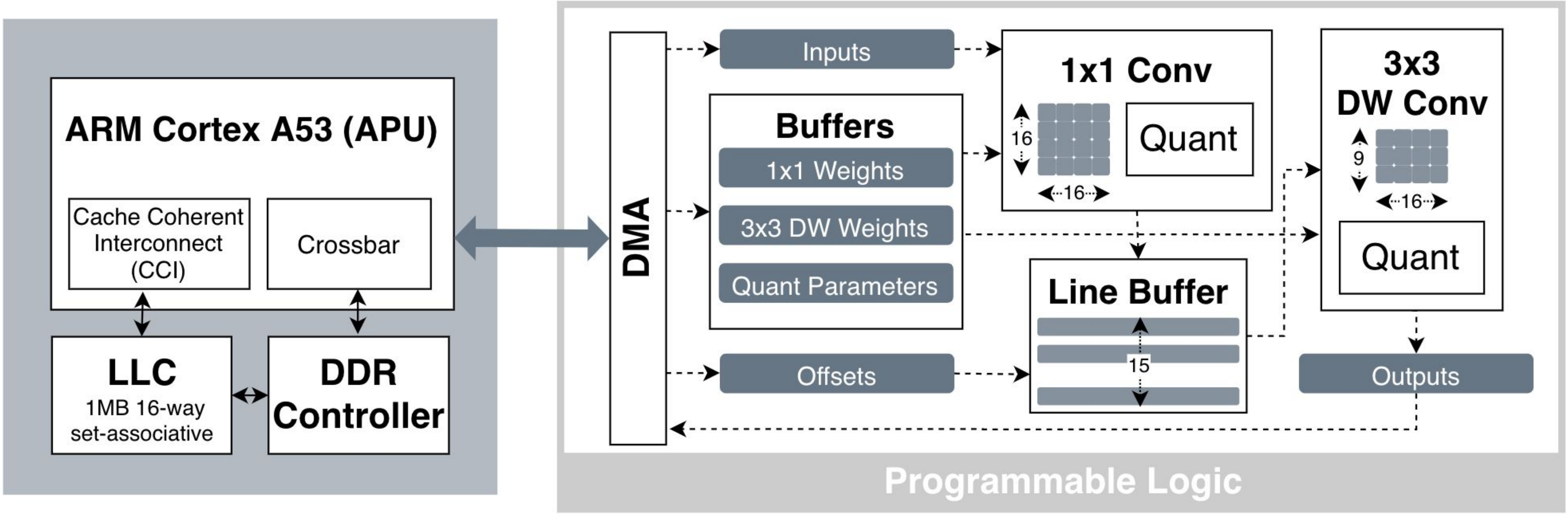
Quantization

1. Quantize the corresponding weights and activations
2. Round and bound the sampling offsets of the deformable convolution

Simplified Operations Types

1. 1x1 convolution
2. 3x3 depthwise (deformable) convolution
3. Quantization
4. Split, shuffle, concatenation

Overall Accelerator Architecture



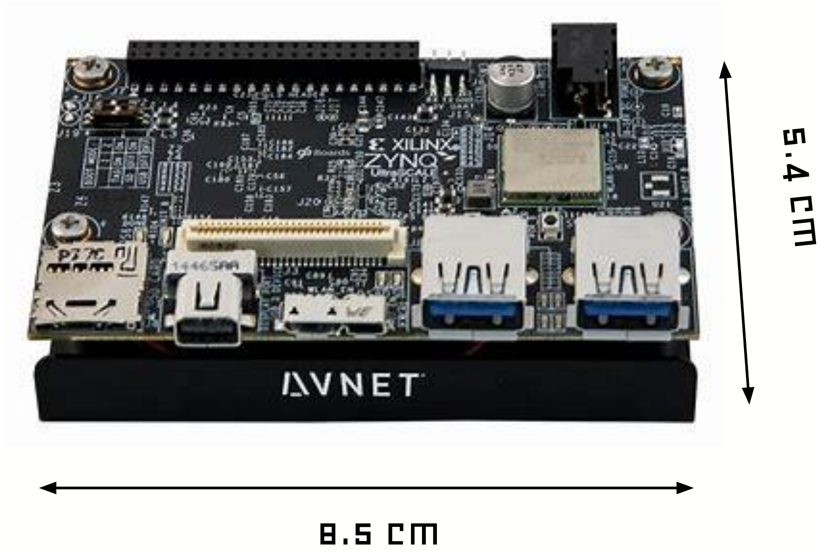
- **Dataflow** architecture for executing a subgraph of *1x1 conv* and *3x3 dw deformable conv*

Outline

- Motivation
- Deformable Convolution
- Operation Codesign
- Detection System Codesign
- Results

Experimental Setup

- Avnet Ultra96 Board (Xilinx ZU3EG FPGA)



- Resource Utilization:

LUT	FF	BRAM	DSP
34144 (48.4%)	41827 (29.6%)	216 (100%)	360 (100%)

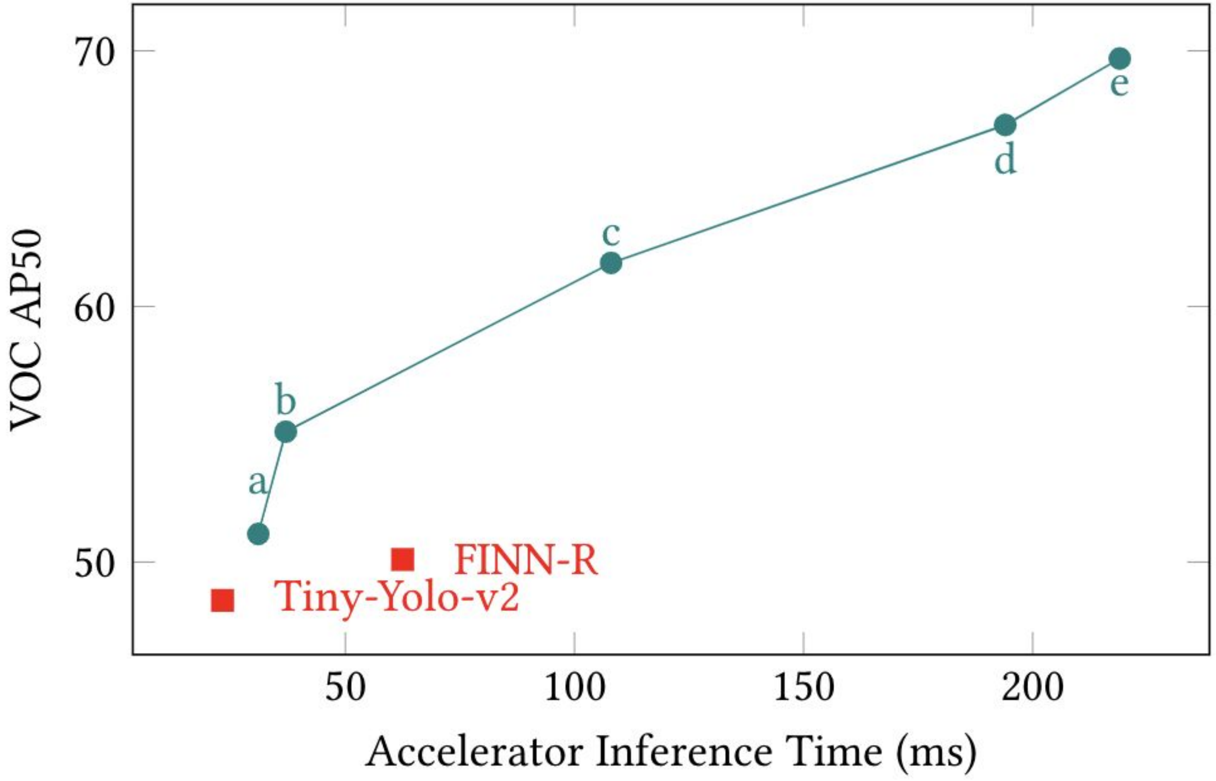
CoDeNet Configurations

Detector	Resolution	DownSample	Weights	Activations	Model Size	MACs	AP50
Tiny-YOLO	416×416	MaxPool	32-bit	32-bit	60.5 MB	3.49 G	57.1
CoDeNet1× (config a)	256×256	Stride4	32-bit	32-bit	6.06 MB	0.29 G	53.0
			4-bit	8-bit	0.76 MB	0.29 G	51.1
CoDeNet1× (config b)	256×256	Stride2+MaxPool	32-bit	32-bit	6.06 MB	0.29 G	57.5
			4-bit	8-bit	0.76 MB	0.29 G	55.1
CoDeNet1× (config c)	512×512	Stride4	32-bit	32-bit	6.06 MB	1.14 G	64.6
			4-bit	8-bit	0.76 MB	1.14 G	61.7
CoDeNet2× (config d)	512×512	Stride4	32-bit	32-bit	23.2 MB	3.54 G	69.6
			4-bit	8-bit	2.90 MB	3.54 G	67.1
CoDeNet2× (config e)	512×512	Stride2+MaxPool	32-bit	32-bit	23.2 MB	3.58 G	72.4
			4-bit	8-bit	2.90 MB	3.58 G	69.7

- Higher accuracy
- 10x smaller without quantization, 79.6x smaller with quantization

Accuracy-Latency Tradeoff

- 4-bit weights, 8-bit activations
- Batch size 1
- On VOC dataset



Berkeley
UNIVERSITY OF CALIFORNIA

Results

	Platform	Input Resolution	Framerate (fps)	Test Dataset	Precision	Accuracy
Finn-R [2] [28]	Ultra96	-	16	VOC07	w1a3	AP50(50.1)
Tiny-Yolo-v2 [11]	Zynq-706 XC7Z045	224 × 224	43.1		w16a16	AP50(48.5)
Ours (config a)		256 × 256	32.2			AP50(51.1)
Ours (config b)		256 × 256	26.9			AP50(55.1)
Ours (config c)	Ultra96	512 × 512	9.3	VOC07	w4a8	AP50(61.7)
Ours (config d)		512 × 512	5.2			AP50(67.1)
Ours (config e)		512 × 512	4.6			AP50(69.7)

- Our accelerator both achieves high accuracy (AP50(55.1)) and good framerate (26.9FPS).

Berkeley
UNIVERSITY OF CALIFORNIA

Results

	Test Dataset	Precision	Accuracy	Framerate (fps)
FINN-R[1]	VOC07	w1a3	AP50(50.1)	16
Ours (a)	VOC07	w4a8	AP50(51.1)	32.2
Ours (b)			AP50(55.1)	26.9
Ours (c)			AP50(61.7)	9.3
Ours (d)			AP50(67.1)	5.2
Ours (e)			AP50(69.7)	4.6

- Our accelerator both achieves high accuracy (AP50(55.1)) and good framerate (26.9FPS).

Berkeley UNIVERSITY OF CALIFORNIA

1 Blott, Michaela, et al. "FINN-R: An end-to-end deep-learning framework for fast exploration of quantized neural networks." *ACM Transactions on Reconfigurable Technology and Systems (TRETS)*

Conclusion

Our codesigned input-adaptive object detection pipeline can achieve both high accuracy and good efficiency on embedded FPGAs

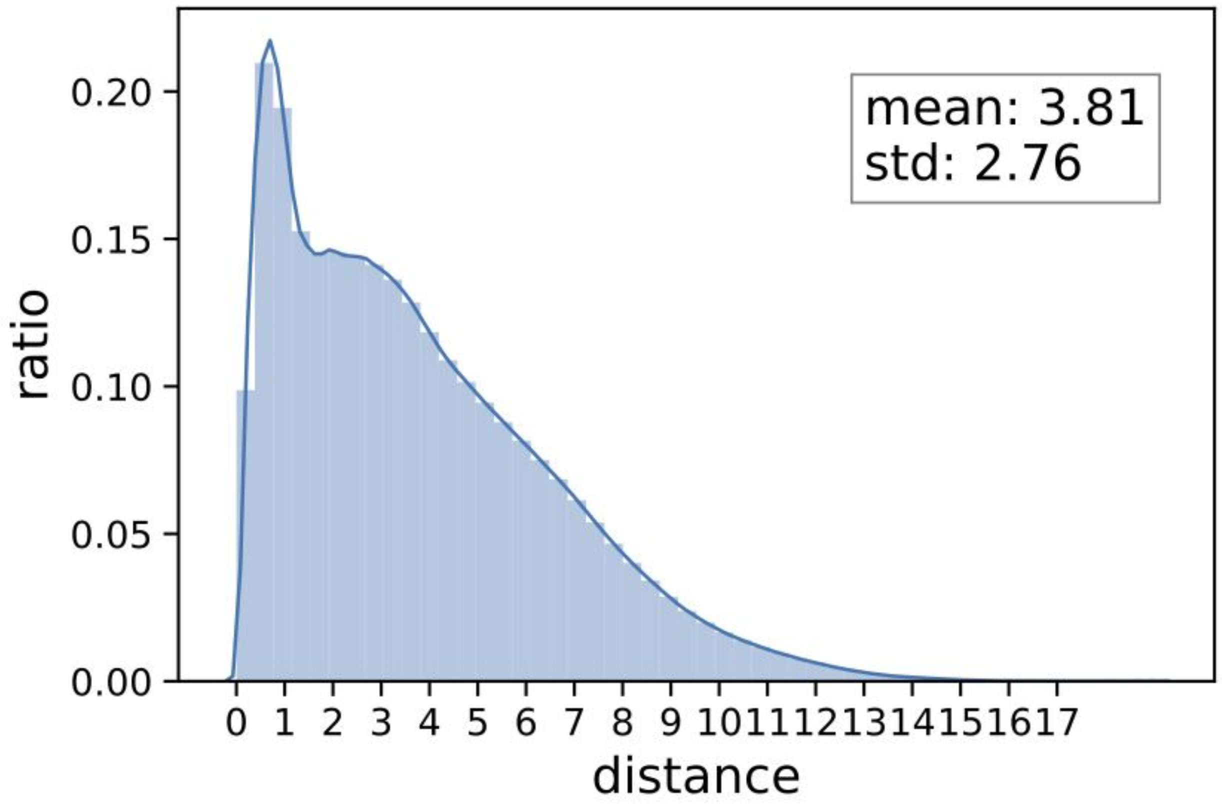
Questions?

Email: qijing.huang@berkeley.edu

Access to code:

https://github.com/hqjenny/CoDeNet

Example Sample Distance



Distance Distribution on 5000 images from COCO