Algorithm-Hardware Co-design for Deformable Convolution

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

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

Berkeley
UNIVERSITY OF CALIFORNIA

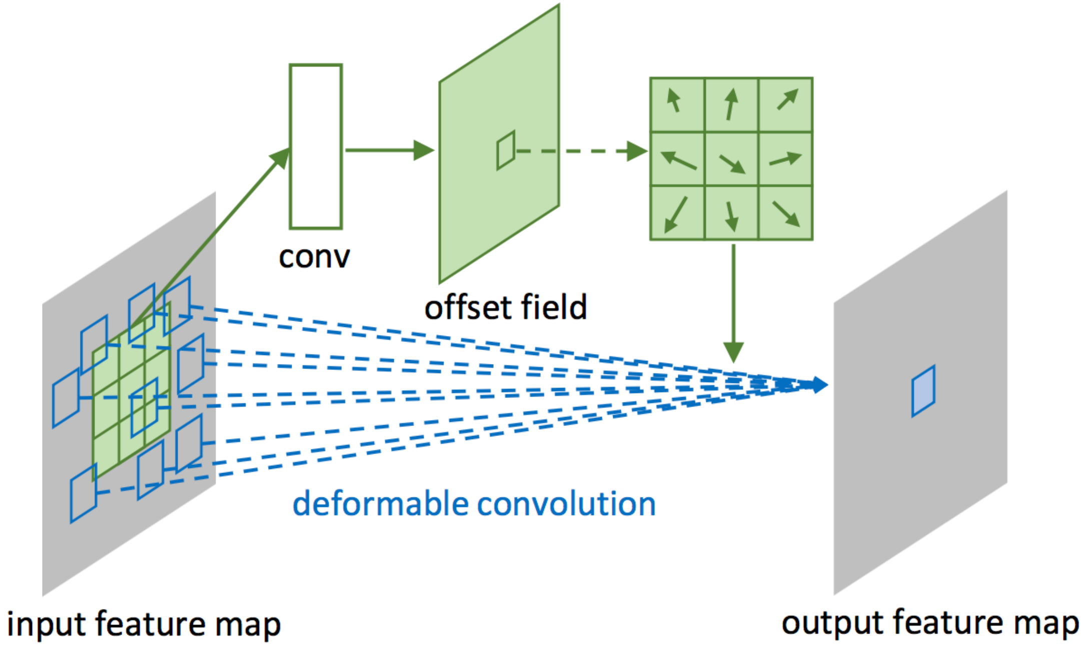
EMC2 Workshop @ NeurIPS 2019

Motivation

- **Deformable Convolution** is an input-adaptive dynamic operation that samples inputs from variable spatial locations
- Its sampling locations vary with:
  - Different input images
  - Different output pixel locations
- It captures the spatial variance of objects with different:
  - Scales
  - Aspect Ratios
  - Rotation Angles
- Challenges:
  - Increased compute and memory requirements
  - Irregular input-dependent memory access patterns
    - Not friendly for dataflows that leverage the spatial reuse

Motivation

- **Deformable Convolution** is an input-adaptive dynamic operation that samples inputs from variable spatial locations
- Its sampling locations vary with:
  - Different in
  - Different ou
- It captures th
  - Scales
  - Aspect Rat
  - Rotation Ar
- Challenges:
  - Increasec
  - Irregular i
    - Not frie



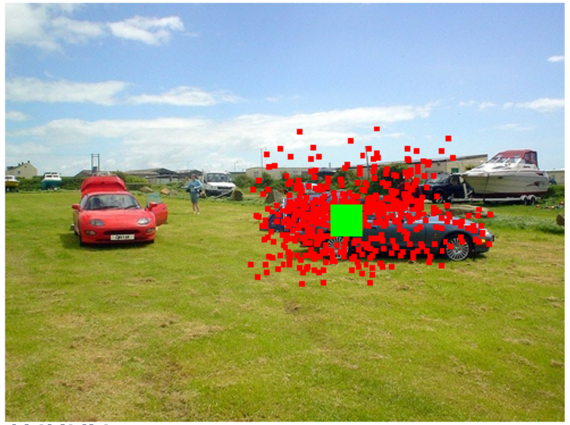
1. Generate offsets
2. Sample from input feature map

Motivation

- **Deformable Convolution** is an input-adaptive dynamic operation that samples inputs from variable spatial locations
- Its sampling locations vary with:
  - Different input images
  - Different output pixel locations
- It captures the spatial variance of objects with different:
  - Scales
  - Aspect Ratios
  - Rotation Angles
- Challenges:
  - Increased compute and memory requirements
  - Irregular input-dependent memory access patterns
    - Not friendly for dataflows that leverage the spatial reuse

Motivation

- **Deformable Convolution** is an input-adaptive dynamic operation that samples inputs from variable spatial locations
- Its sampling locations vary with:
  - Different input images
  - Different output pixel locations
- It captures the spatial variance of objects with different:
  - Scales
  - Aspect Ratic
  - Rotation Ang
- Challenges:
  - Increased uireme
  - Irregular in ccess p
    - Not frien he spatial



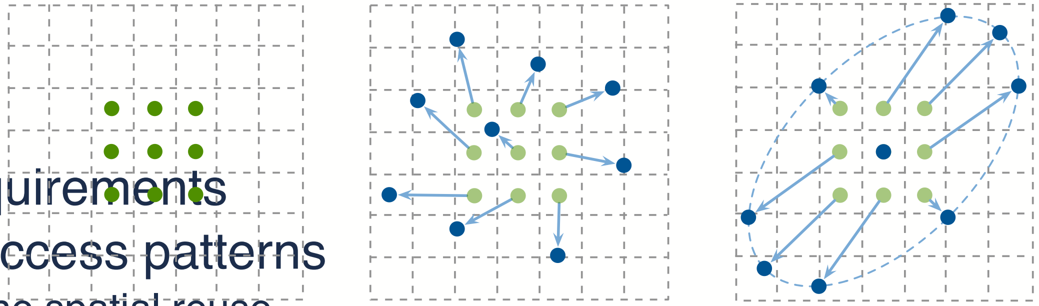
Sampling Locations (in red) for Different Output Pixels (in green)

Motivation

- **Deformable Convolution** is an input-adaptive dynamic operation that samples inputs from variable spatial locations
- Its sampling locations vary with:
  - Different input images
  - Different output pixel locations
- It captures the spatial variance of objects with different:
  - Scales
  - Aspect Ratios
  - Rotation Angles
- Challenges:
  - Increased compute and memory requirements
  - Irregular input-dependent memory access patterns
    - Not friendly for dataflows that leverage the spatial reuse

Motivation

- **Deformable Convolution** is an input-adaptive dynamic operation that samples inputs from variable spatial locations
- Its sampling locations vary with:
  - Different input images
  - Different output pixel locations
- It captures the spatial variance of objects with different:
  - Scales
  - Aspect Ratios
  - Rotation Angles
- Challenges:
  - Increased compute and memory requirements
  - Irregular input-dependent memory access patterns
    - Not friendly for dataflows that leverage the spatial reuse



Variable Receptive Fields

Berkeley
UNIVERSITY OF CALIFORNIA

Motivation

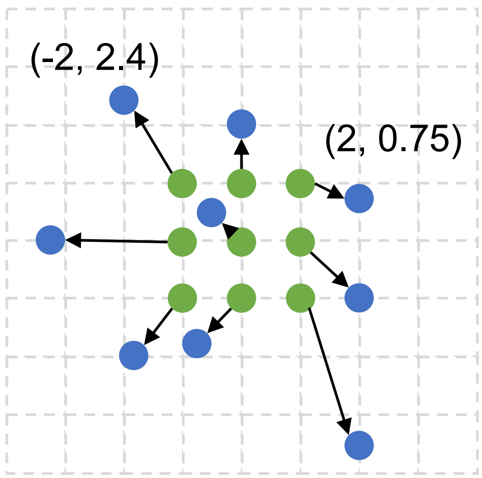
- **Deformable Convolution** is an input-adaptive dynamic operation that samples inputs from variable spatial locations
- Its sampling locations vary with:
  - Different input images
  - Different output pixel locations
- It captures the spatial variance of objects with different:
  - Scales
  - Aspect Ratios
  - Rotation Angles
- Challenges:
  - Increased compute and memory requirements
  - Irregular input-dependent memory access patterns
    - Not friendly for dataflows that leverage the spatial reuse

Motivation

- Why codesign algorithm and hardware?
  - **Inefficient Model Designs** – many CV tasks use large inefficient models and operations solely optimized for accuracy
  - **Limited Hardware Resources** – embedded devices have limited compute resources and a strict energy and power budgets
  - **Real-time Requirements** – accelerators must guarantee response within certain time constraints
- Goals: codesign **algorithms** and **accelerators** that *satisfy embedded system constraints and fall on the pareto curve of the accuracy-latency tradeoff.*

Algorithm-Hardware Codesign

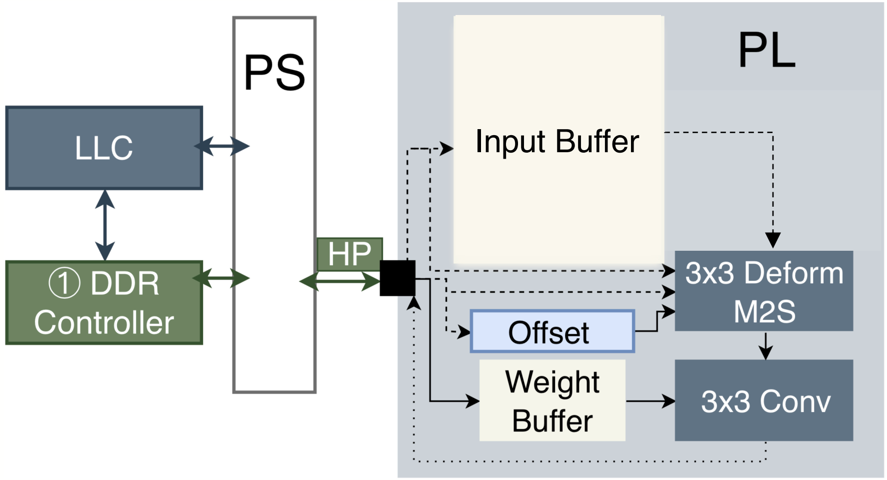
Algorithm Modification:



0. Original Deformable

Accuracy [1](mIoU ↑): **79.9**

Hardware Optimization:

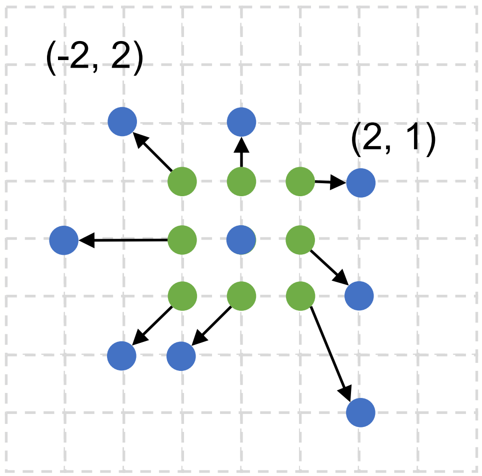


- **Preloads weights to on-chip buffer**
- **Loads input and offsets directly from DRAM**

[1] Accuracy for Semantic Segmentation on CityScapes

Algorithm-Hardware Codesign

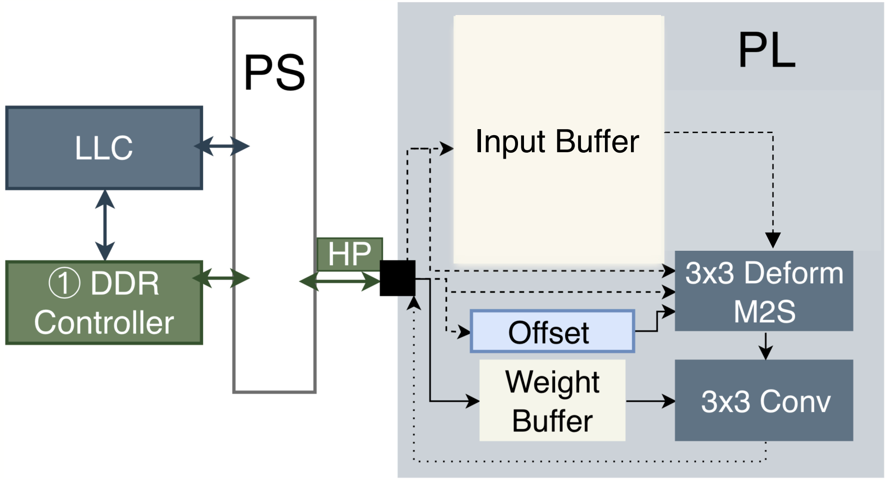
Algorithm Modification:



1. Rounded Offsets

Accuracy [1](mIoU ↑): **79.6** **↓ 0.3**

Hardware Optimization:

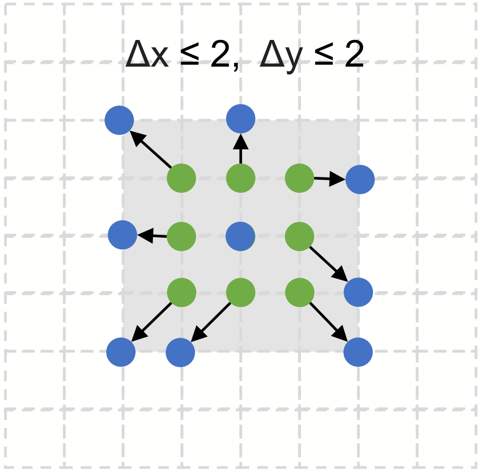


- **Reduces the computation for bilinear interpolation**

[1] Accuracy for Semantic Segmentation on CityScapes

Algorithm-Hardware Codesign

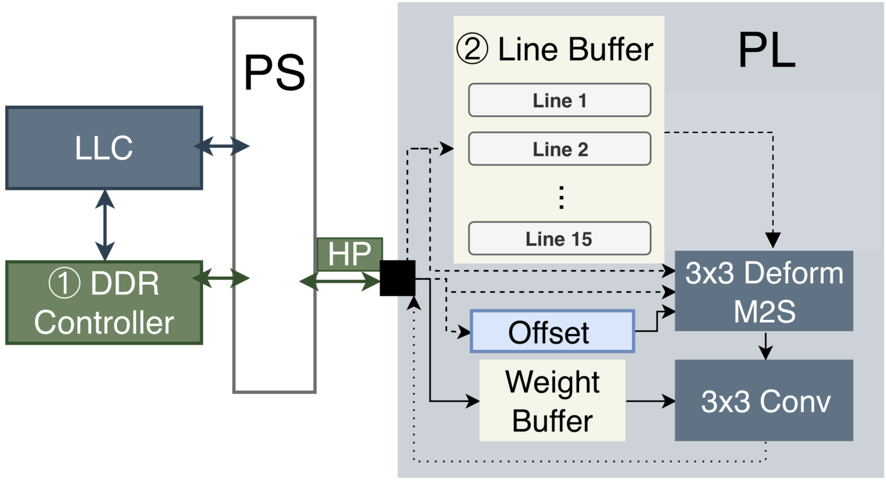
Algorithm Modification:



2. Bounded Range

Accuracy [1](mIoU ↑): **79.4** **↓ 0.2**

Hardware Optimization:

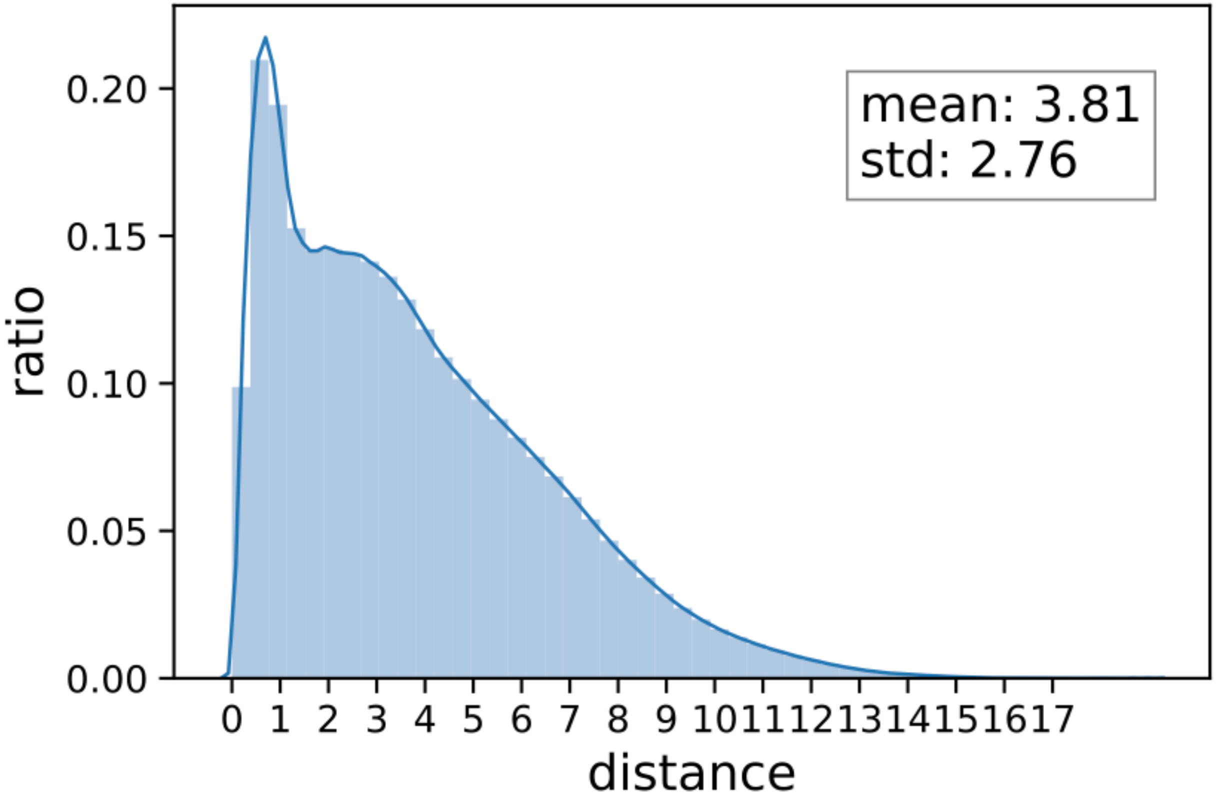


- **Buffers inputs in the on-chip line buffer to allow spatial reuse**

[1] Accuracy for Semantic Segmentation on CityScapes

Berkeley UNIVERSITY OF CALIFORNIA

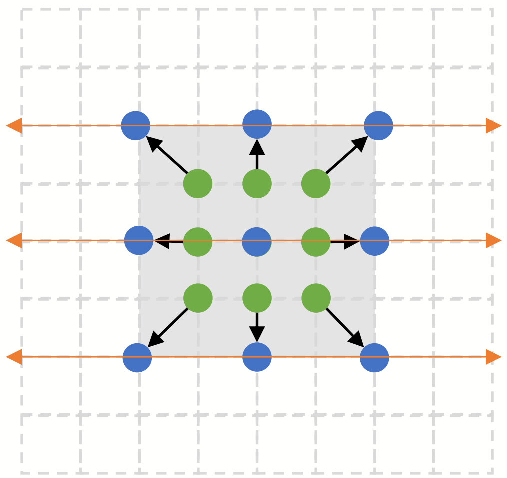
Example Sample Distance



Distance Distribution on 5000 images from COCO

Algorithm-Hardware Codesign

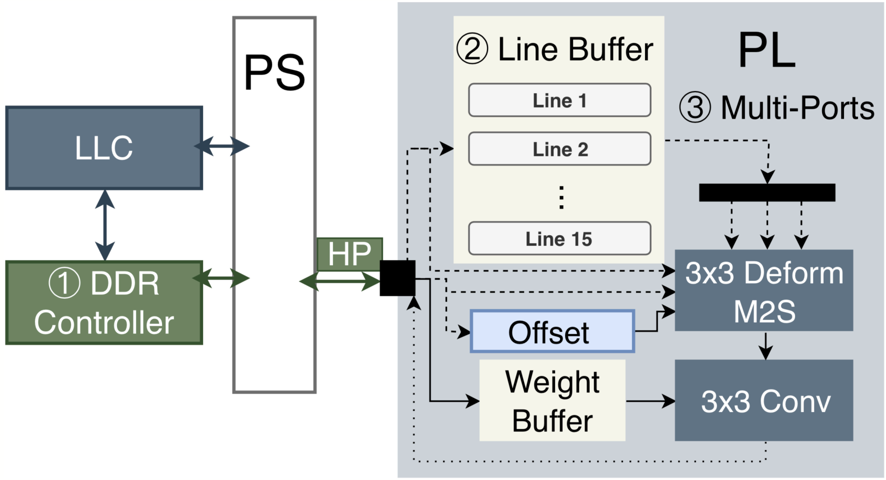
Algorithm Modification:



3. Rectangular Shape

Accuracy [1](mIoU ↑): **78.7** **↓ 0.7**

Hardware Optimization:



- **Improves on-chip memory bandwidth**

[1] Accuracy for Semantic Segmentation on CityScapes

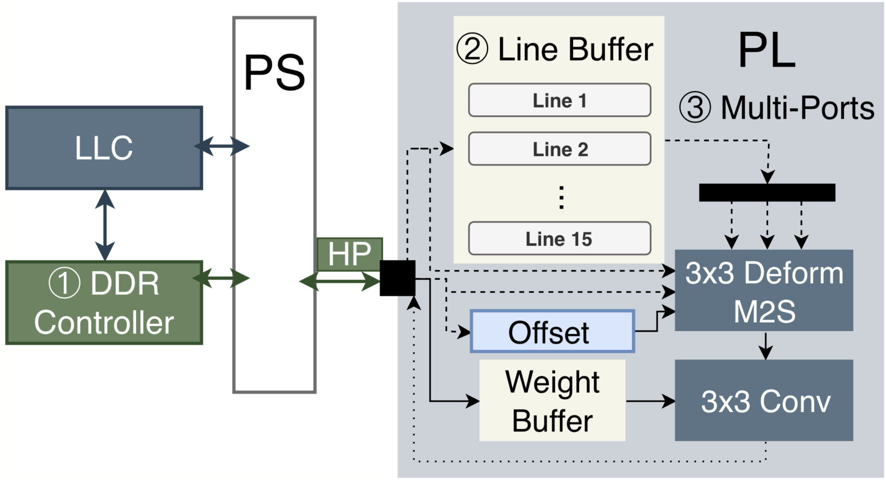
Algorithm-Hardware Codesign

Algorithm Modification:

4. Efficient Feature Extractor
5. Depthwise Convolution

Feature Extractor	Operation	mIoU ↑
DLA	DeformConv	79.9
ShuffleNetV2	DeformConv	70.1
ShuffleNetV2	DeformConv + Depthwise	68.0

Hardware Optimization:



- **Reduce the total MACs**

[1] Accuracy for Semantic Segmentation on CityScapes

Results

Hardware Performance

Operation	Original	Deformable	Bound (buffered)	Square (multi-ported)	Without LLC		With LLC	
					Latency (ms)	GOPs	Latency (ms)	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

- Our algorithm-hardware co-design methodology for the deformable convolution achieves a **1.36×** and **9.76×** speedup respectively for the *full* deformable convolution and *depthwise* deformable convolution on FPGA

Email: qijing.huang@berkeley.edu

[1] Accuracy for Semantic Segmentation on CityScapes