IEEE ISCAS Ph.D. Forum

ISCAS 2024

Johnny Rhe
Sungkyunkwan Univ.
djwhsdj@skku.edu

Advisor: Dr. Jong Hwan Ko
https://iris.skku.edu/

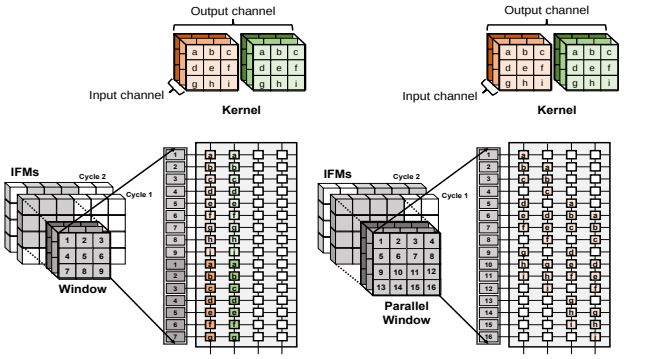
Optimizing Convolutional Weight Mapping for Energy-Efficient In-Memory CNN Inference

Research Abstract

- **In-Memory Computing (IMC)** architectures are increasingly being used for the convolutional neural networks (CNNs) inference
- Mapping method is important to reduce computing cycles for minimizing the energy consumption
- This research aims to develop **mapping-aware optimization for energy-efficient CNN inference in IMC architecture** through algorithmic, network architectures and their training innovations

Background

- **Convolutional Weight Mapping Methods**



❒ **Image to Column (im2col)**
- Basic mapping (Fig.1 (a))
- 3D kernel to array column

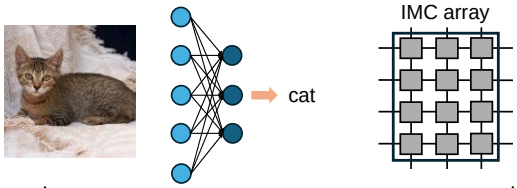
❒ **Shift and Duplicate Kernel (SDK)**
- Advanced mapping method
- Deployment of duplicated kernels

- SDK mapping improves utilization and reduces cycles
→ However, it still faces limitations, including **limited utilization enhancement** and **irregular weight deployment**

Motivation

§ **Recent networks**
- **Weights**: 1 Mega ~ 150 M params
- **Inputs**: 32 x 32 ~ 1024 x 1024 (~1K pixels) (~1M pixels)

§ **Recent single array**
- **Size:** 32x32 (~1K cells) ~512 x 512 (~0.26 M cells)



How to map?

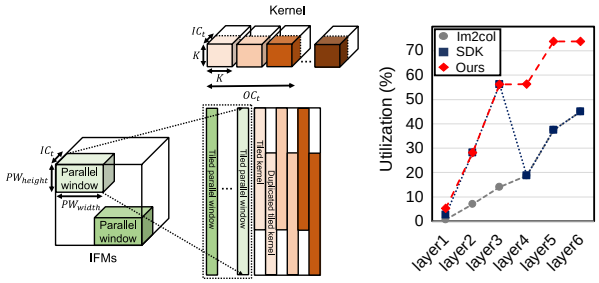
§ **Determines 100~100,000 computing cycles**
- Energy consumption
- Inference latency

❒ **Limitation on Recently Used Single Array Sizes**
- Computing cycle is determined by mapping method
- **Unable to map the entire weight matrix** of convolutional layer
→ A new mapping method is needed to minimize cycles

❒ **Need for Weight Compression for Mapping Efficiency**
- **# weight params.** ≫ **# memory cells** in single IMC array
- Limited efficiency when using only mapping methods
→ New mapping-aware weight compression is required

Proposed Method & Results

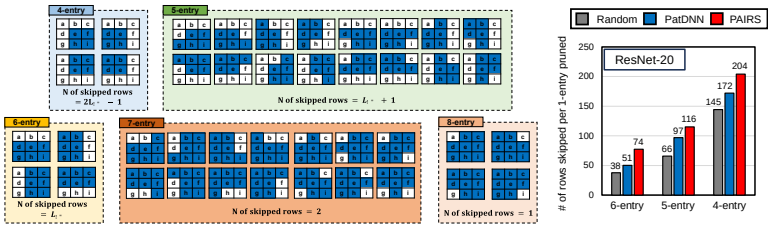
1. Novel Mapping Method



❒ Various Window SDK (VW-SDK) [DATE 2022]
- **Dividing the channels into several tiles (channel tiling)**
→ Deploying duplicated kernels in limited array sizes
- Various 2D shapes of the parallel window
→ **Extended search space** of PW shapes

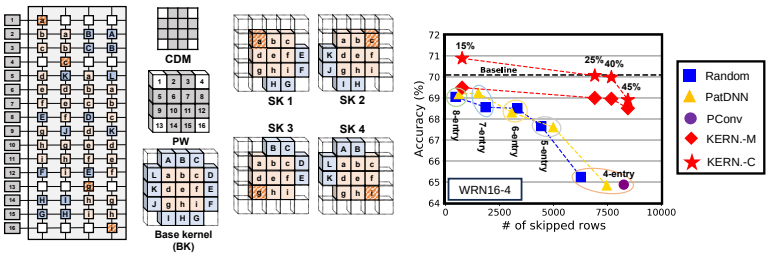
2. Weight Compression w/ Weight Pruning

❒ Variable Windows & Channels SDK [JETCAS 2022]
- VW-SDK + channel pruning
- Removes some channels that lead to low array utilization
→ **Further optimize the VW-SDK mapping method**



❒ Pruning-Aided Row-Skipping (PAIRS) [ISLPED 2023]
- Row-skipping via pattern-based pruning to compress weight matrix
- **SDK mapping-aware pruning pattern design**
→ Maximum weight matrix compression rate

3. Weight Compression w/o Weight Pruning



❒ Kernel Shape Control (KERNTROL)
- KERN.-M: **weight omission** technique [ICCAD 2023]
→ **Prevents pruning the important weight element**

- KERN.-C: KERN.-M + **compensatory weights** [TCAS-I 2024]
→ **100% array utilization + minimize accuracy drop**

Author's Info.

- **9+3** International Conference papers (**DATE** 2022, **ISLPED** 2023, **ICCAD** 2023)
- **3+1** International Journal papers (**TCAS-I** 2021, 2024, **JETCAS** 2022, 2024)
- **Most Popular Poster Award at ASP-DAC 2024**