State-of-the-Art Model Quantization and Optimization for Efficient Edge AI

Hyunjin Kim

Compiler Team Lead

DEEPX AI

DEEPX

DEEPX: Edge AI Solution

NPU Technology

NPU based SoC Technology

AI HW/SW Optimization

Developing one of the most efficient NPU technologies

Developing SoC ASIC with NPU for commercial products

Optimizing both AI HW & SW to provide the highest NPU efficiency

DEEPX: Edge AI Solution

NPU Technology

NPU based SoC Technology

AI HW/SW Optimization

Developing one of the most efficient NPU technol

This Talk!

Developing SoC ASIC with NPU for commercial products

Optimizing both AI HW & SW to provide the highest NPU efficiency

DEEPX Mission

- Satisfy the **EDGE** AI inference requirement
- EDGE AI inference requirement
  - Maintain high accuracy with quantization
  - Support various AI models (operation coverage)
  - Minimize HW Cost: SRAM

DEEPX Mission

- Satisfy the **EDGE** AI inference requirement
- EDGE AI inf
  - Maintain
  - Support v
  - Minimize HW Cost: SRAM

Our Solution:
HW/SW Co-design

High Accuracy with Quantization

	DXNN-Lite	DXNN-Pro	DXNN-Master
Norm. Accuracy Over FP32 (avg.)	99.01%	99.42%	100.54%
Time Cost	~ 1 min	- 1~2 hours	50+ GPU Hours * Training env.

- HW/SW Co-Design
  - Selective quantization to avoid accuracy drop
    - 8-bit quantized ops + FP32 ops

Operation Coverage

- High operation coverage with minimum implementation cost

- HW/SW Co-Design
  - Increase op coverage via combinations of supported ops
  - TransConv: Resize + Padding + Conv
  - DilatedConv: multiple Convs
  - ArgMax: Two-stage implementation

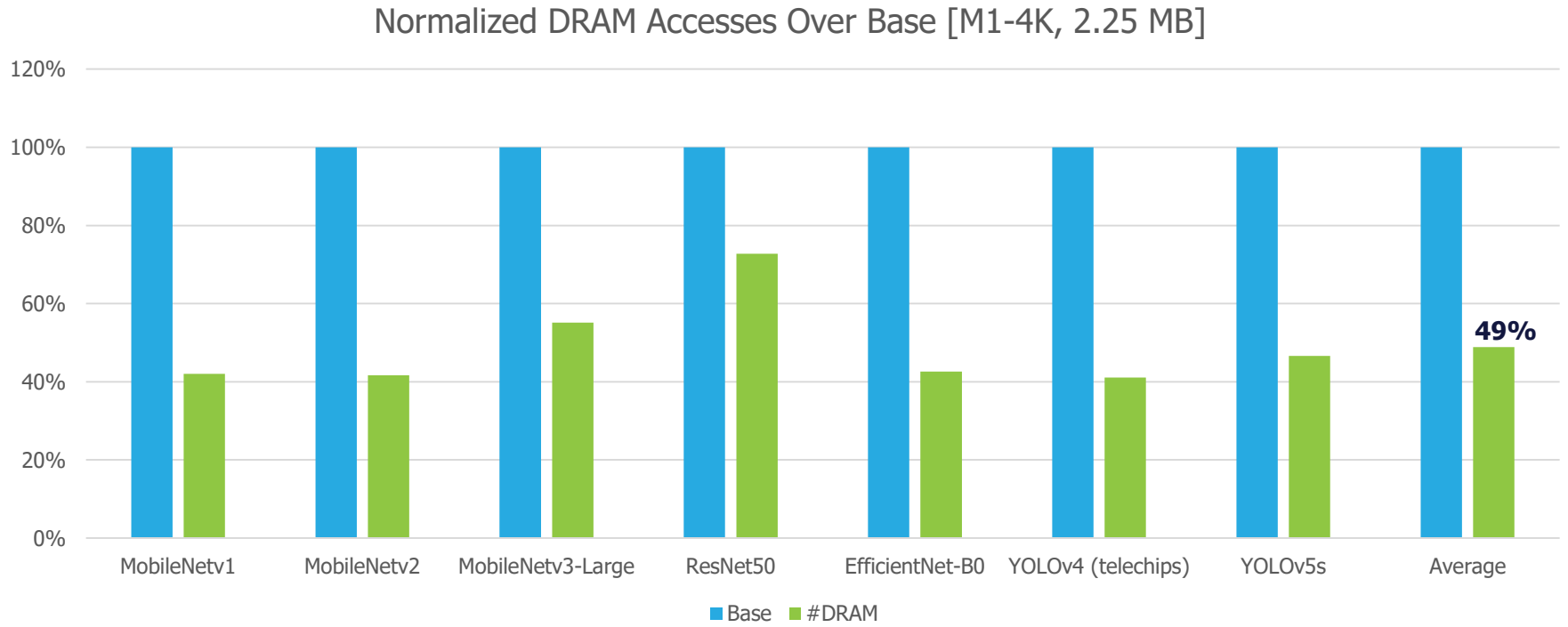
Minimize HW Cost: SRAM

- DEEPX NPUs have small SRAM sizes.

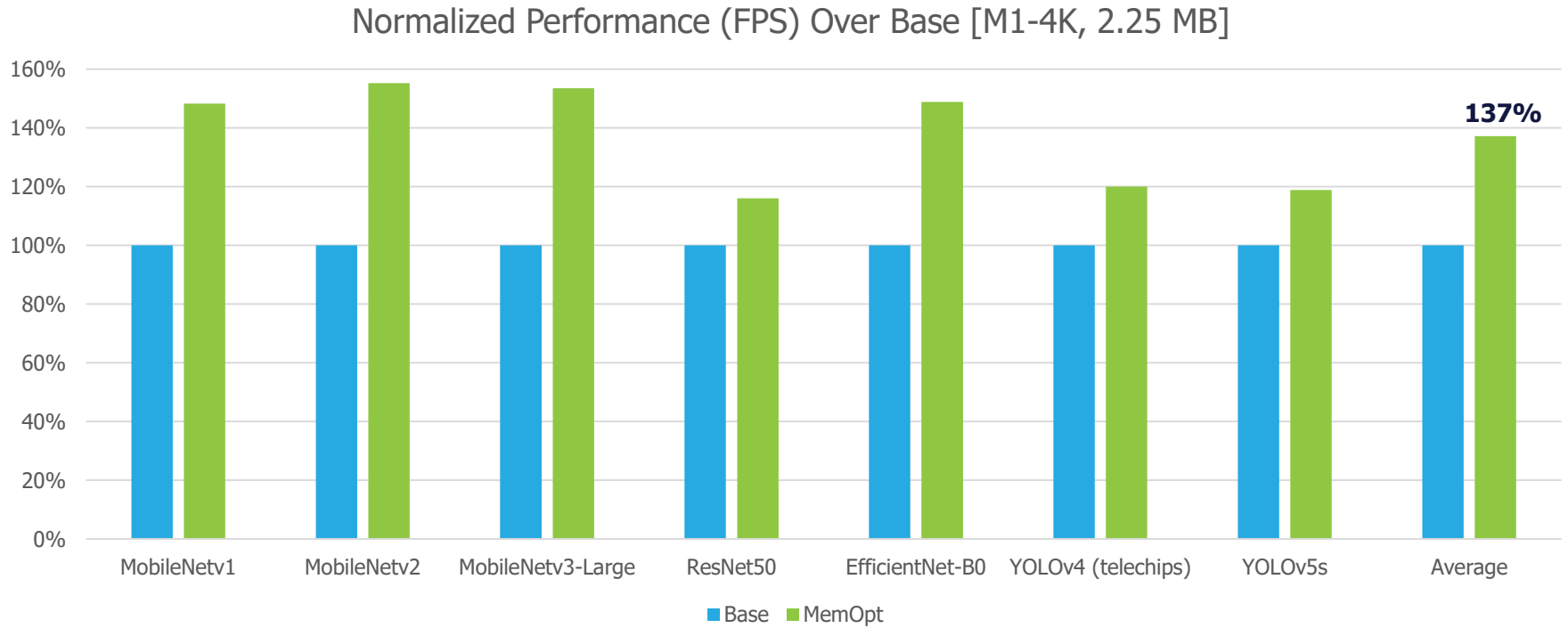
SRAM: 1 MB **SRAM: 2.25 MB** **SRAM: 8.25 MB**

- HW/SW Co-Design
  - DEEPX compiler's memory optimization maximizes SRAM utilization to reduce DRAM accesses

Memory Optimization: DRAM Accesses



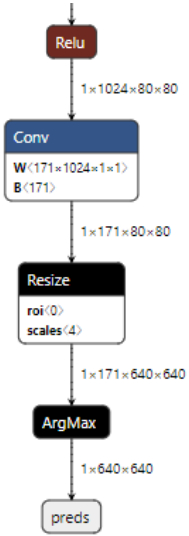
Memory Optimization: Performance



Case Study: Fused Operations

Fused Operation: Resize + Argmax (1/4)

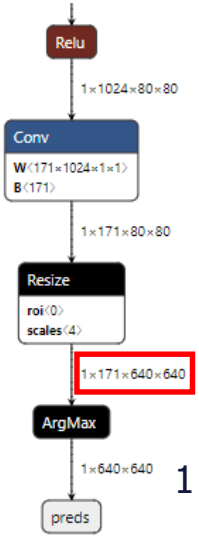
- Segmentation Models
  - `Resize` → `Argmax` at the end for pixel-wise classification



bisenet_v2

Fused Operation: Resize + Argmax (2/4)

- Segmentation Models
  - `Resize` → `Argmax` at the end for pixel-wise classification



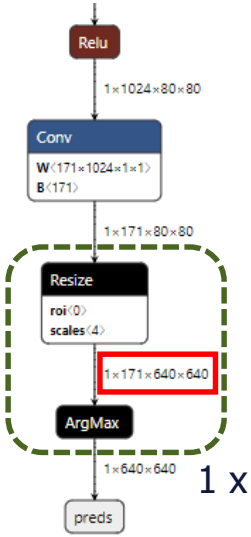
bisenet_v2

1 x 171 x 640 x 640 x 4 (FP32) = 267 MB

1 x 640 x 640 x 2 (INT16) = 800 KB

Fused Operation: Resize + Argmax (3/4)

- Segmentation Models
  - `Resize` → `Argmax` at the end for pixel-wise classification



bisenet_v2

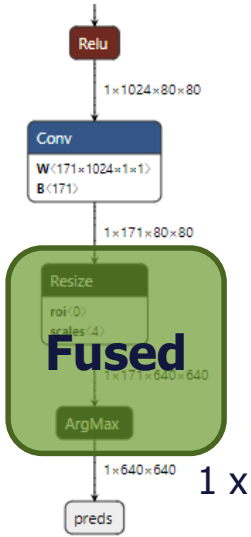
Fusing Resize and Argmax by performing Argmax during Resize via **2-stage Argmax** (171 → 1)

1 x 171 x 640 x 640 x 4 (FP32) = 267 MB

1 x 640 x 640 x 2 (INT16) = 800 KB

Fused Operation: Resize + Argmax (4/4)

- Segmentation Models
  - `Resize` → `Argmax` at the end for pixel-wise classification



bisenet_v2

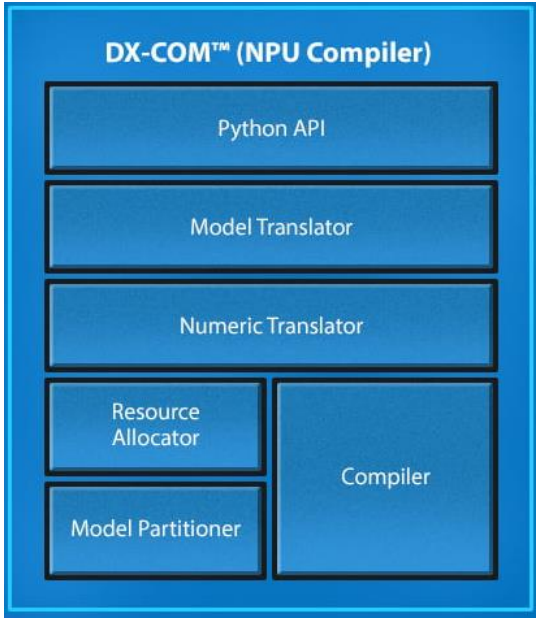
Fusing Resize and Argmax by performing Argmax during Resize via **2-stage Argmax** (171 → 1)

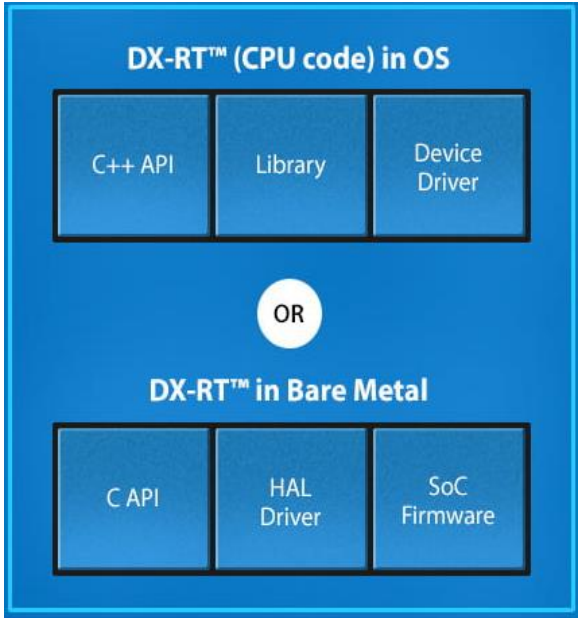
~~1 x 171 x 640 x 640 x 4 (FP32) = 267 MB~~

1 x 640 x 640 x 2 (INT16) = 800 KB

DXNN SDK Tutorial

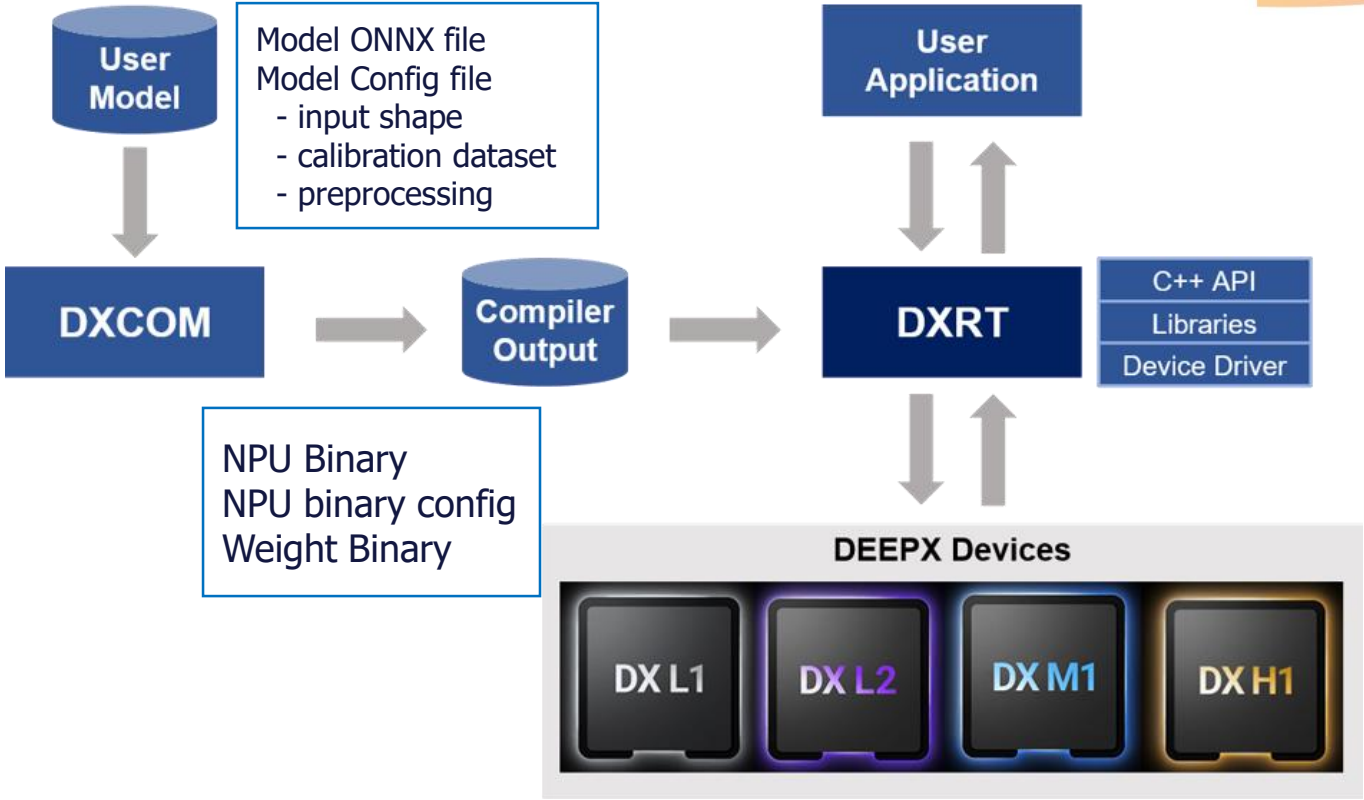
DXNN™ – DEEPX NPU SDK





- **Beta version of DXNN has been released to customers.**

DXNN SDK Tutorial: Overview

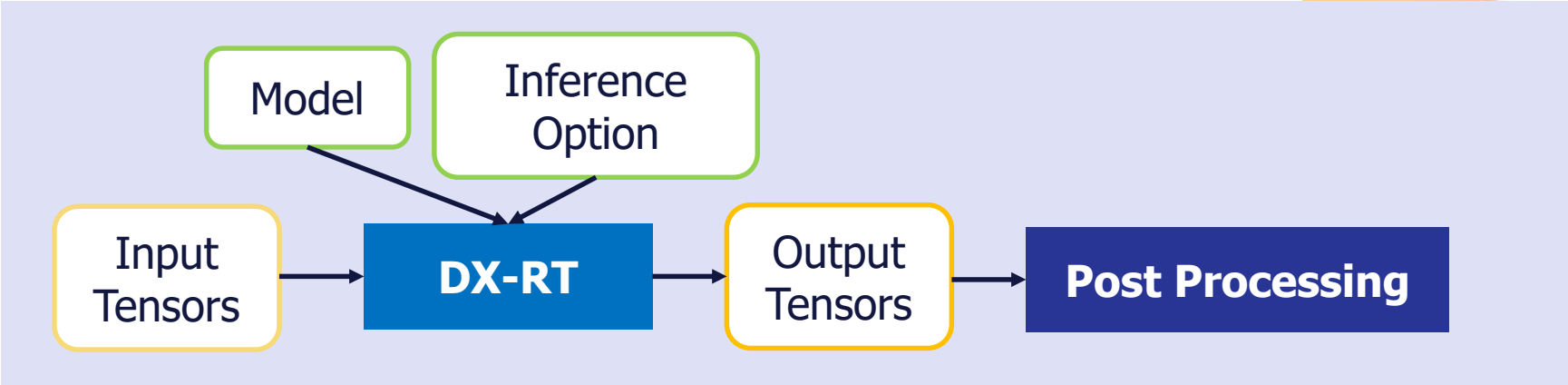


DXNN SDK Tutorial: Model Compilation

- DX-COM compiles the model onxx file
- It also performs the **quantization** as well.
  - The config file includes the model meta data.
    - input shape, input pre-processing parameter, …

```
$./dx_com/dx_com \
    --model_path sample/yolov5s.onnx \
    --config_path sample/yolov5s.json \
    --output_dir output/yolov5s
```

DXNN SDK Tutorial: Run Inference




```
$run_model --model /dxrt/models/yolov5s \  # compilation output directory
           --input /dxrt/models/yolov5s/input/npu_input_0.bin \
           --ref /dxrt/models/yolov5s/output/output_last/npu_output_0.bin \
           --output output.bin
          --loop 5
```

DXNN SDK Tutorial: Post Processing

- User should configure the parameters for post-processing.
  - The parameters are not included in the compilation output.
- Example: yolov5s

```
YoloParam yolov5s_512 = {
    .imgSize = 512,
    .confThreshold = 0.25,
    .scoreThreshold = 0.3,
    .iouThreshold = 0.4,
    .numClasses = 80,
    .layers = {
    …
};
```

DXNN SDK Tutorial: DX-RT API #1

```
#include "dxrt/dxrt_api.h“

#1 Set Inference Options: 1 NPU, 1 WORKER THREAD, 2 MEMORY BUFFER
dxrt::InferenceOption option(1, 1, 2, dxrt::InferenceMode::MODE_SYNC);
```

DXNN SDK Tutorial: DX-RT API #2

```
#include "dxrt/dxrt_api.h“

#1 Set Inference Options: 1 NPU, 1 WORKER THREAD, 2 MEMORY BUFFER
dxrt::InferenceOption option(1, 1, 2, dxrt::InferenceMode::MODE_SYNC);

auto ie = dxrt::InferenceEngine("yolov5s", &option);  #2 Load Model
```

DXNN SDK Tutorial: DX-RT API #3

```
#include "dxrt/dxrt_api.h“

#1 Set Inference Options: 1 NPU, 1 WORKER THREAD, 2 MEMORY BUFFER
dxrt::InferenceOption option(1, 1, 2, dxrt::InferenceMode::MODE_SYNC);

auto ie = dxrt::InferenceEngine("yolov5s", &option);  #2 Load Model

auto inputs = ie.GetInput();  #3 Set inputs
```

DXNN SDK Tutorial: DX-RT API #4

```
#include "dxrt/dxrt_api.h“

#1 Set Inference Options: 1 NPU, 1 WORKER THREAD, 2 MEMORY BUFFER
dxrt::InferenceOption option(1, 1, 2, dxrt::InferenceMode::MODE_SYNC);

auto ie = dxrt::InferenceEngine("yolov5s", &option);  #2 Load Model

auto inputs = ie.GetInput();  #3 Set inputs

auto outputs = ie.Run();  #4 Run Inference
```

DXNN SDK Tutorial: DX-RT API #5

```
#include "dxrt/dxrt_api.h“

#1 Set Inference Options: 1 NPU, 1 WORKER THREAD, 2 MEMORY BUFFER
dxrt::InferenceOption option(1, 1, 2, dxrt::InferenceMode::MODE_SYNC);

auto ie = dxrt::InferenceEngine("yolov5s", &option);  #2 Load Model

auto inputs = ie.GetInput();  #3 Set inputs

auto outputs = ie.Run();  #4 Run Inference

auto result = yolo.PostProc(outputs);  #5 Perform Post Processing
```

DXNN SDK Tutorial: Result

Conclusion

- DEEPX's HW/SW codesign leads to
  - Meeting the required specifications with minimum HW cost
    - Selective quantization
    - The combination of supported ops
    - Minimize HW Cost: SRAM
    - Case Study
      - Reduced DRAM access by fused operation

Resources

1. Demo Booth: #103
2. DEEPX Developers Page: https://deepx.ai/developers
3. Linkedin & Youtube

2023 Embedded Vision Summit

Previous Talks from DEEPX

1. "Toward the Era of AI Everywhere" (Lokwon Kim, CEO)
2. "DEEPX's New M1 NPU Delivers Flexibility, Accuracy, Efficiency and Performance" (Jay Kim, Vice President)