FPGA2018
@Monterey

Tokyo Tech

A Lightweight YOLOv2:
A Binarized CNN with a Parallel Support Vector Regression for an FPGA

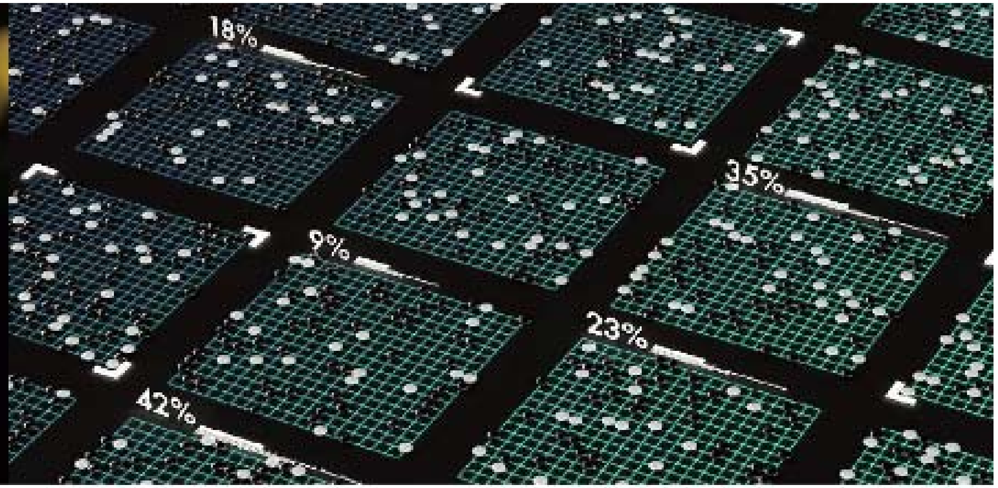
Hiroki Nakahara, Haruyoshi Yonekawa, Tomoya Fujii, Shimpei Sato

Tokyo Institute of Technology, Japan

Outline

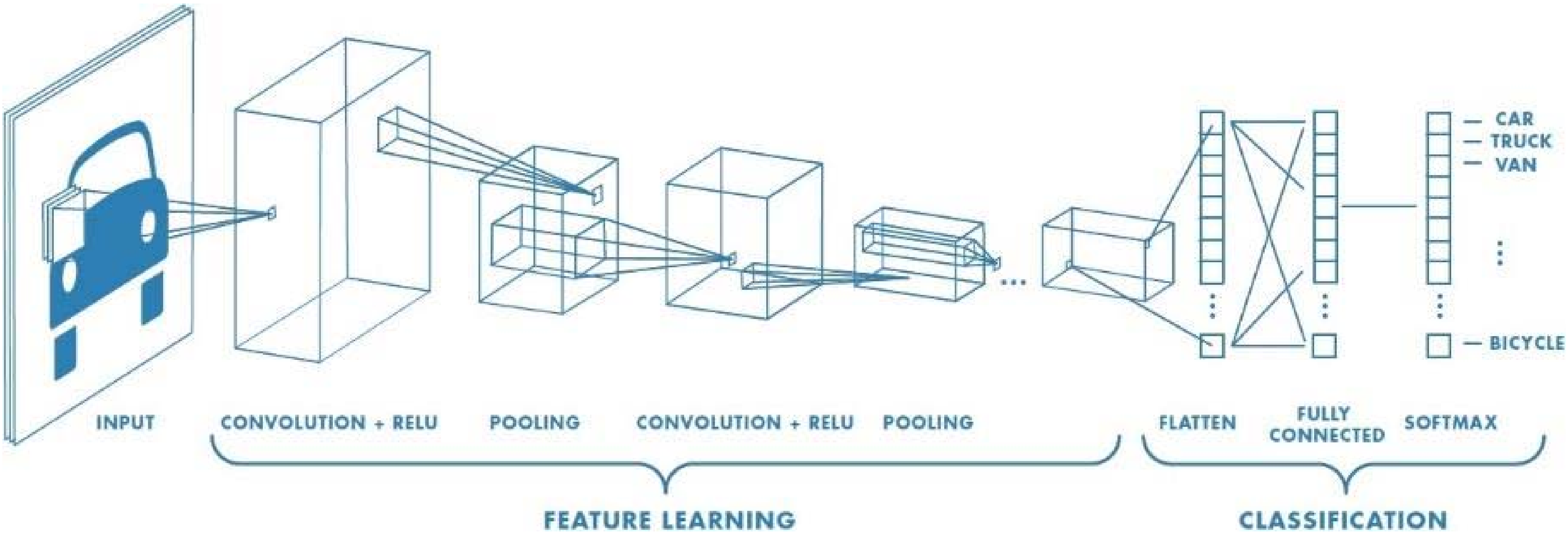
- Background
  - Convolutional Neural Network (CNN)
- Mixed-precision CNN for a Lightweight YOLOv2
  - Binary precision CNN
  - Half precision support vector regression (SVR)
- FPGA Implementation
- Experimental Results
- Conclusion

Deep Learning is Everywhere



Convolutional Neural Network (CNN)

- Convolutional + Fully connected + Pooling
- State-of-the-art performance in an image recognition task
- Widely applicable



Source: https://www.mathworks.com/discovery/convolutional-neural-network.html

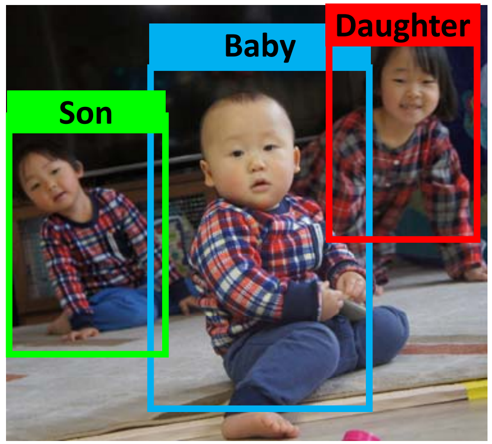
Image Recognition Tasks

Easy

- **Classification**
  - **Answer “category” of the object in an image**

Baby (44%)
Son (23%)
Daughter (33%)

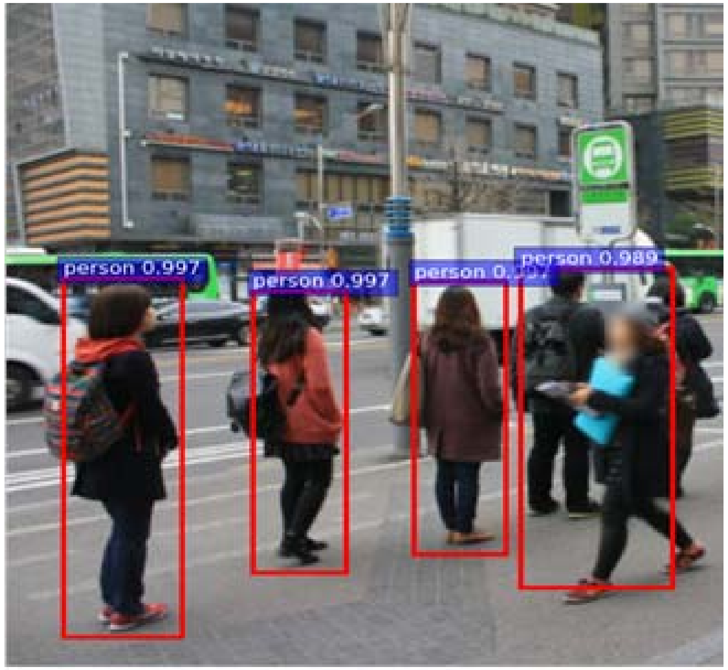
- **<u>Object Detection</u>**
  - **Classification + localization**



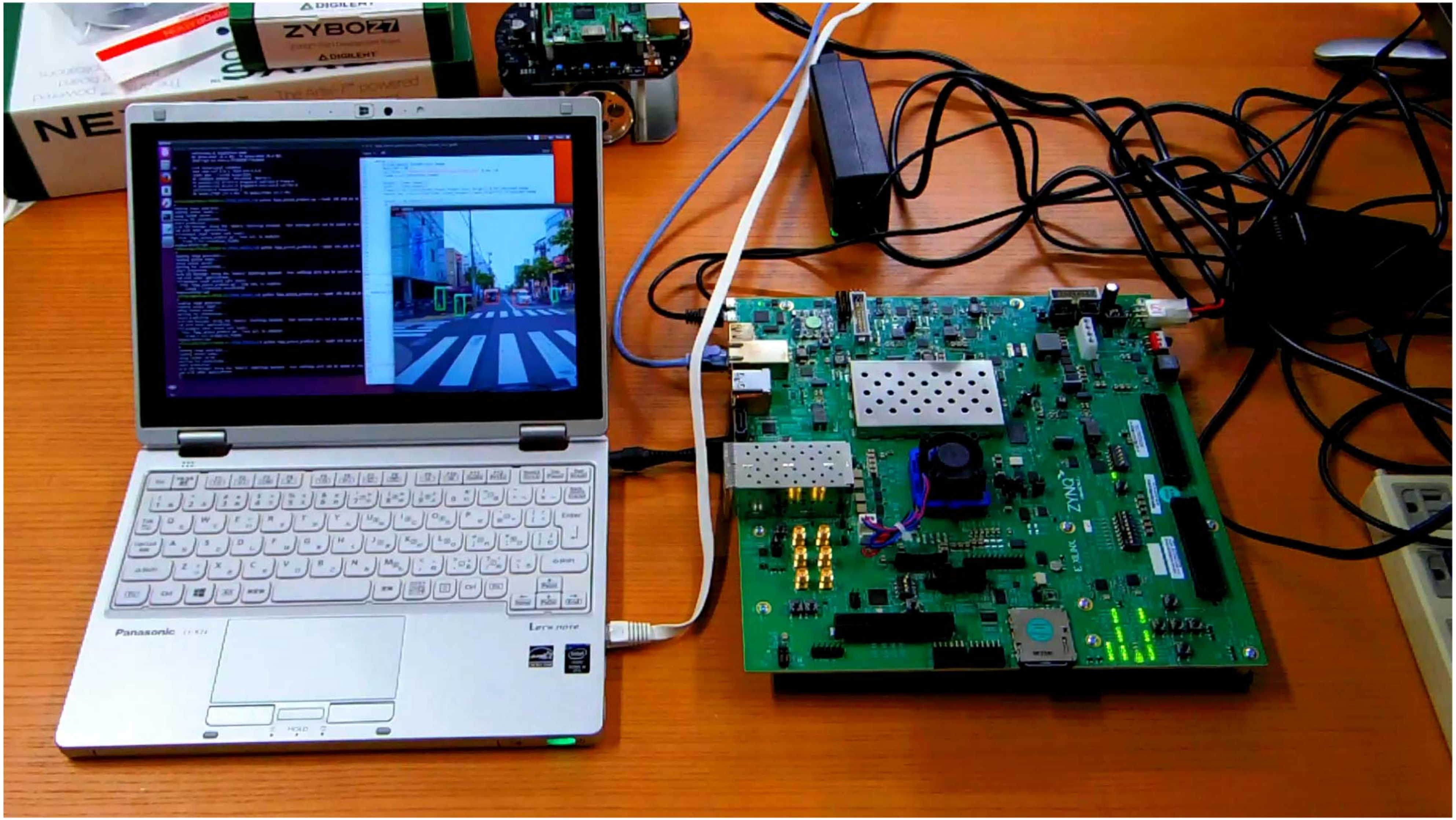
Hard

Applications

- **Robotics, autonomous driving, security, drones…**



Demo



Available at https://www.youtube.com/watch?v=_iMboyu8iWc

Requirements in Embedded System

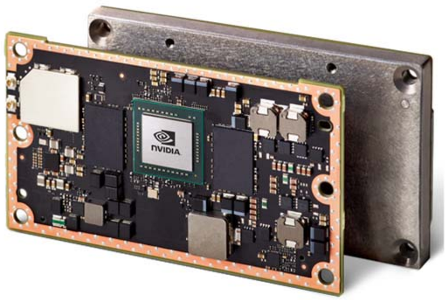
Cloud	Embedded
Many classes (1000s)	Few classes (<10)
Large workloads	Frame rates (15-30 FPS)
High efficiency (Performance/W)	Low cost & low power (1W-5W)
Server form factor	Custom form factor

J. Freeman (Intel), "FPGA Acceleration in the era of high level design", HEART2017

Deep Learning Inference Device

- Flexibility: R&D costs for keeping on evolving algorithms
- Power performance efficiency
- FPGA has flexibility&better performance

CPU
(Raspberry Pi3)



GPU
(Jetson TX2)

FPGA
(UltraZed)

ASIC
(Movidius)

Flexibility ← → **Power Performance Efficiency**

Outline

- Background
  - Convolutional Neural Network (CNN)
- Mixed-precision CNN for a Lightweight YOLOv2
  - Binary precision CNN
  - Half precision support vector regression (SVR)
- FPGA Implementation
- Experimental Results
- Conclusion

Object Detection Problem

- Detecting and classifying multiple objects at the same time
- Evaluation criteria (from Pascal VOC):

Ground truth annotation

Detection results:
>50% overlap of bounding box(BBox) with ground truth
One BBox for each object
Confidence value for each object

$$precision = \frac{\#\,Correct\ detect.}{\#all\ detect.}$$

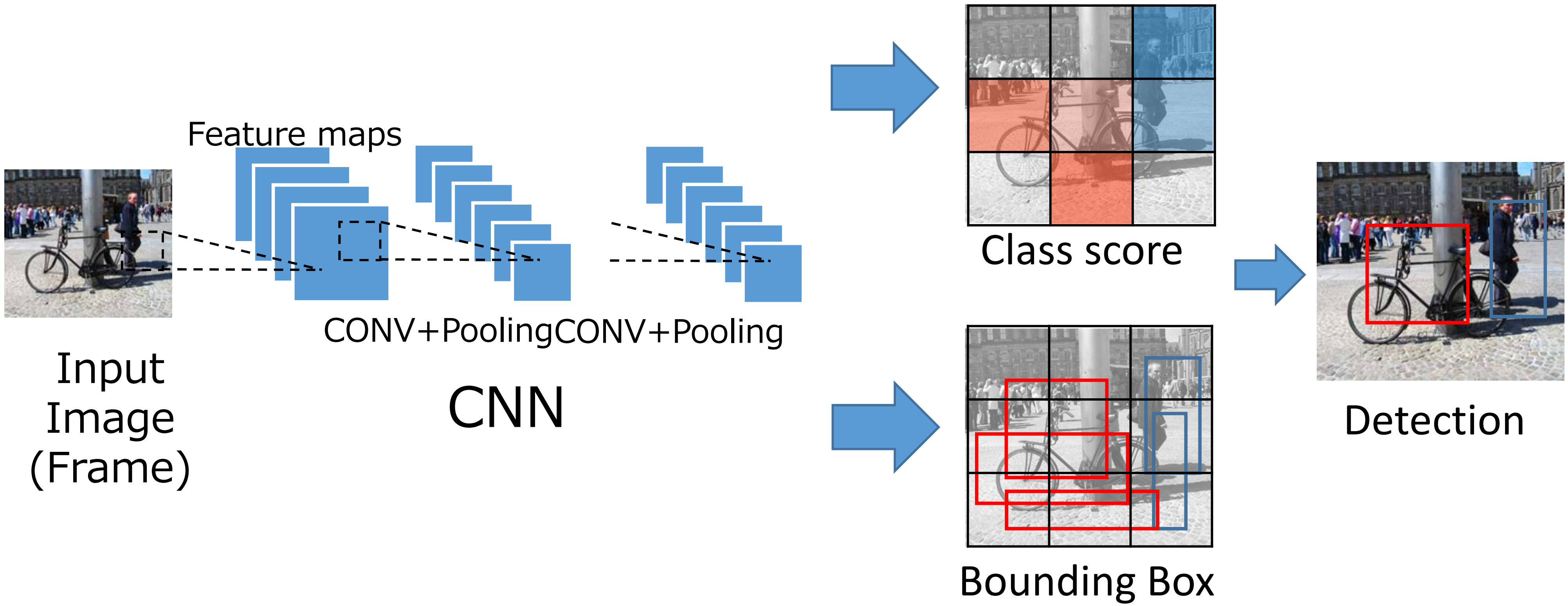
$$recall = \frac{\#\,Correct\ detect.}{\#all\ objects}$$

Average Precision (AP):

$$AP = \frac{1}{11}\sum P_{interp(r), r\in\{0,.1,\ldots,11\}}$$

YOLOv2
(You Only Look Once version 2)

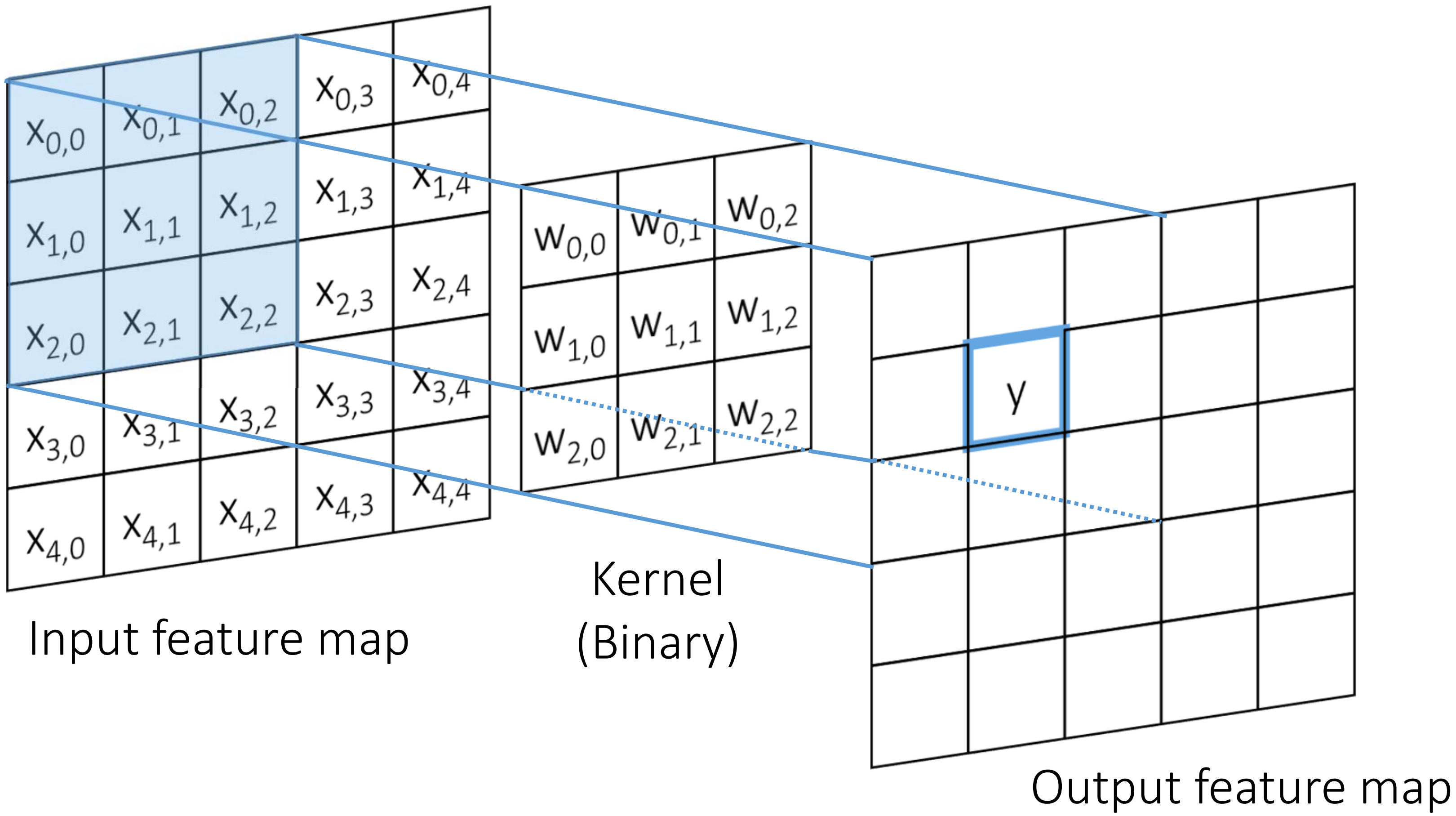
- **Single CNN (One-shot) object detector**
  - **Both a classification and a BBox estimation for each grid**



J. Redmon and A. Farhadi, "YOLO9000: Better, Faster, Stronger," *arXiv preprint arXiv:1612.08242*, 2016.

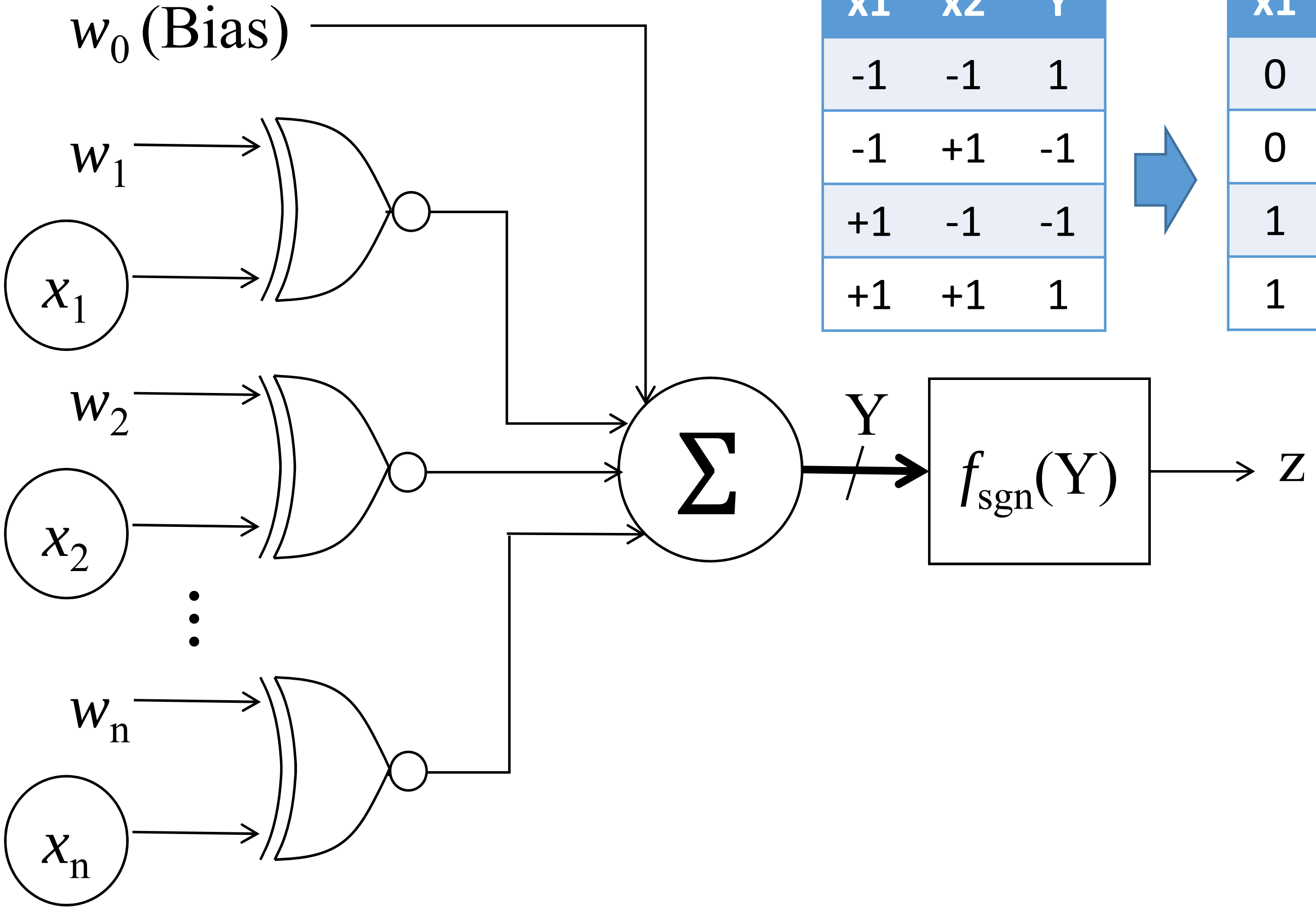
2D Convolutional Operation

- **Computational intensive part of the YOLOv2**



$$
\begin{array}{r}
X_{0,0} \times W_{0,0} \\
X_{0,1} \times W_{0,1} \\
X_{0,2} \times W_{0,2} \\
X_{1,0} \times W_{1,0} \\
X_{1,1} \times W_{1,1} \\
X_{1,2} \times W_{1,2} \\
X_{2,0} \times W_{2,0} \\
X_{2,1} \times W_{2,1} \\
+)\ X_{2,2} \times W_{2,2} \\
\hline
y
\end{array}
$$

Binarized CNN

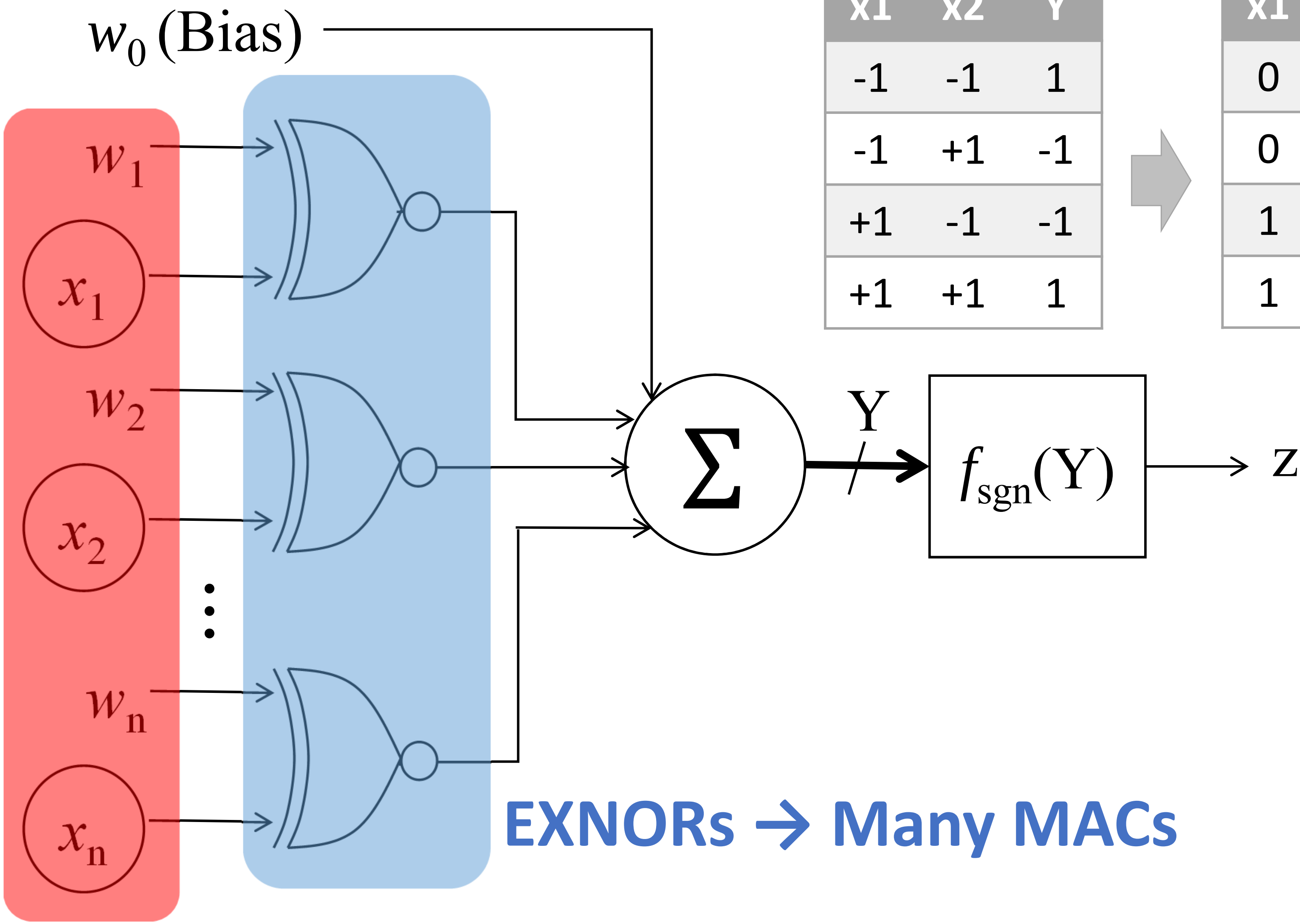


x1	x2	Y
-1	-1	1
-1	+1	-1
+1	-1	-1
+1	+1	1

x1	x2	Y
0	0	1
0	1	0
1	0	0
1	1	1

M. Courbariaux, I. Hubara, D. Soudry, R.E.Yaniv, Y. Bengio, "Binarized neural networks: Training deep neural networks with weights and activations constrained to +1 or -1," *Computer Research Repository (CoRR)*, Mar., 2016, http://arxiv.org/pdf/1602.02830v3.pdf

Improvements by Binarization



x1	x2	Y
-1	-1	1
-1	+1	-1
+1	-1	-1
+1	+1	1

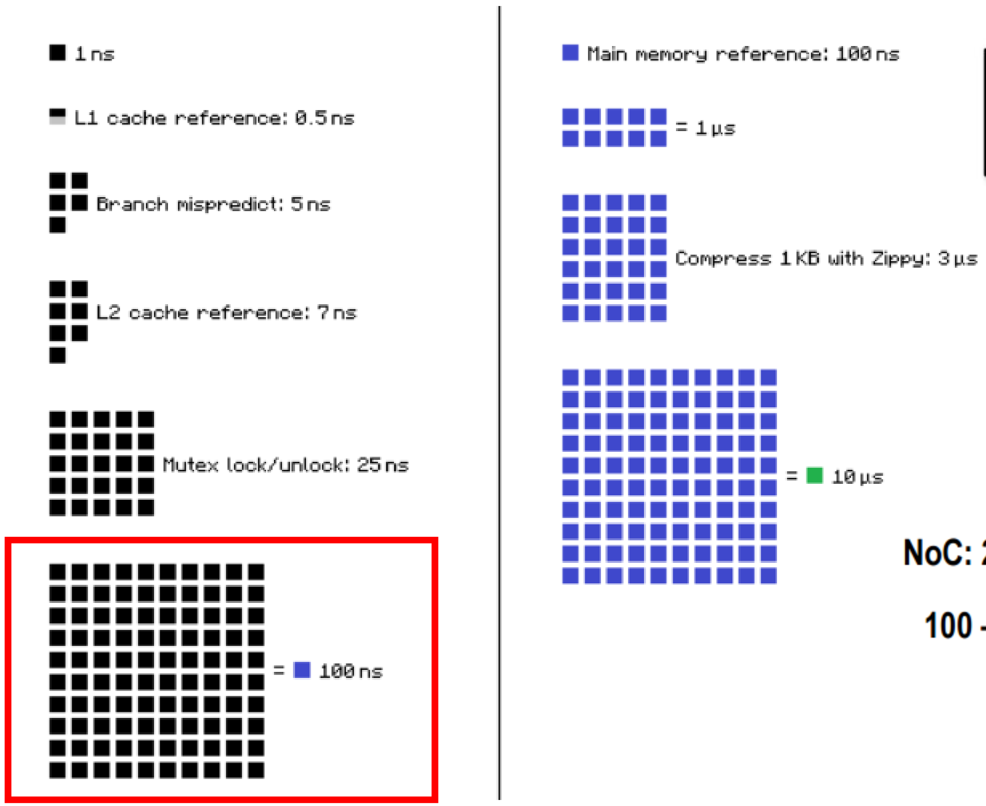
x1	x2	Y
0	0	1
0	1	0
1	0	0
1	1	1

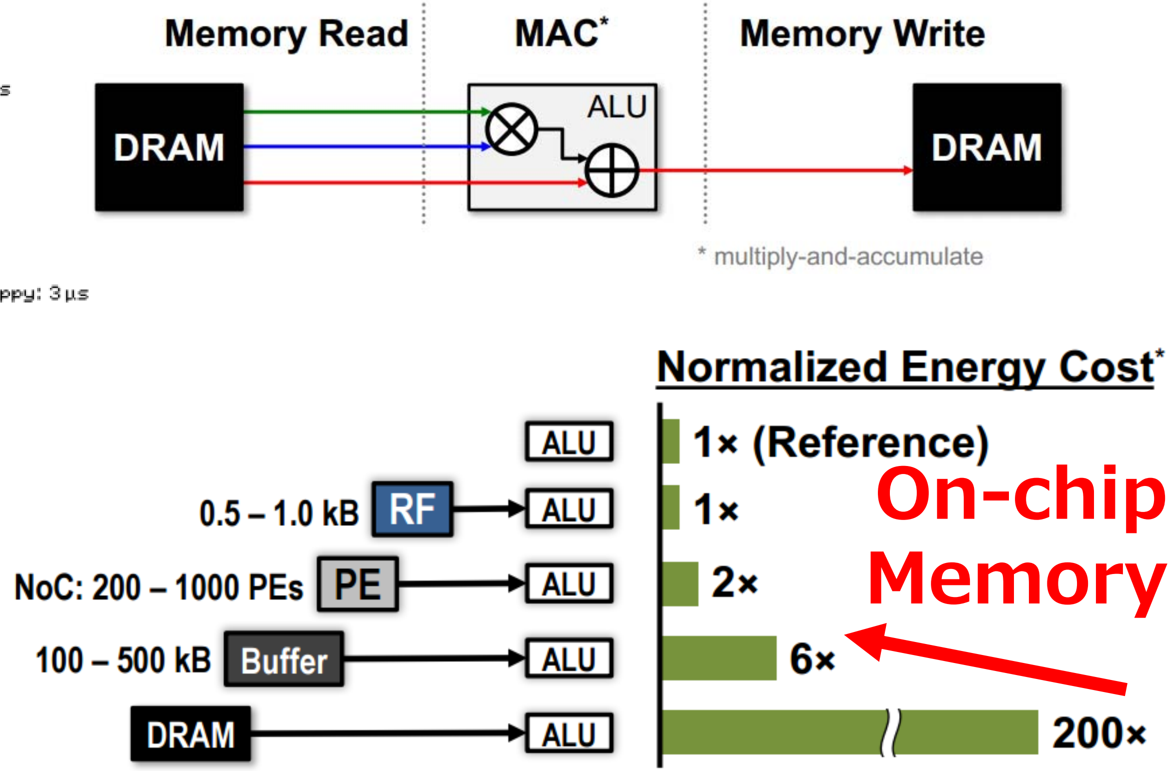
EXNORs → Many MACs

Binary Precision → On-chip Memory

Near Memory Realization by Binarization

- High bandwidth (Left)
- Less power consumption (Right)

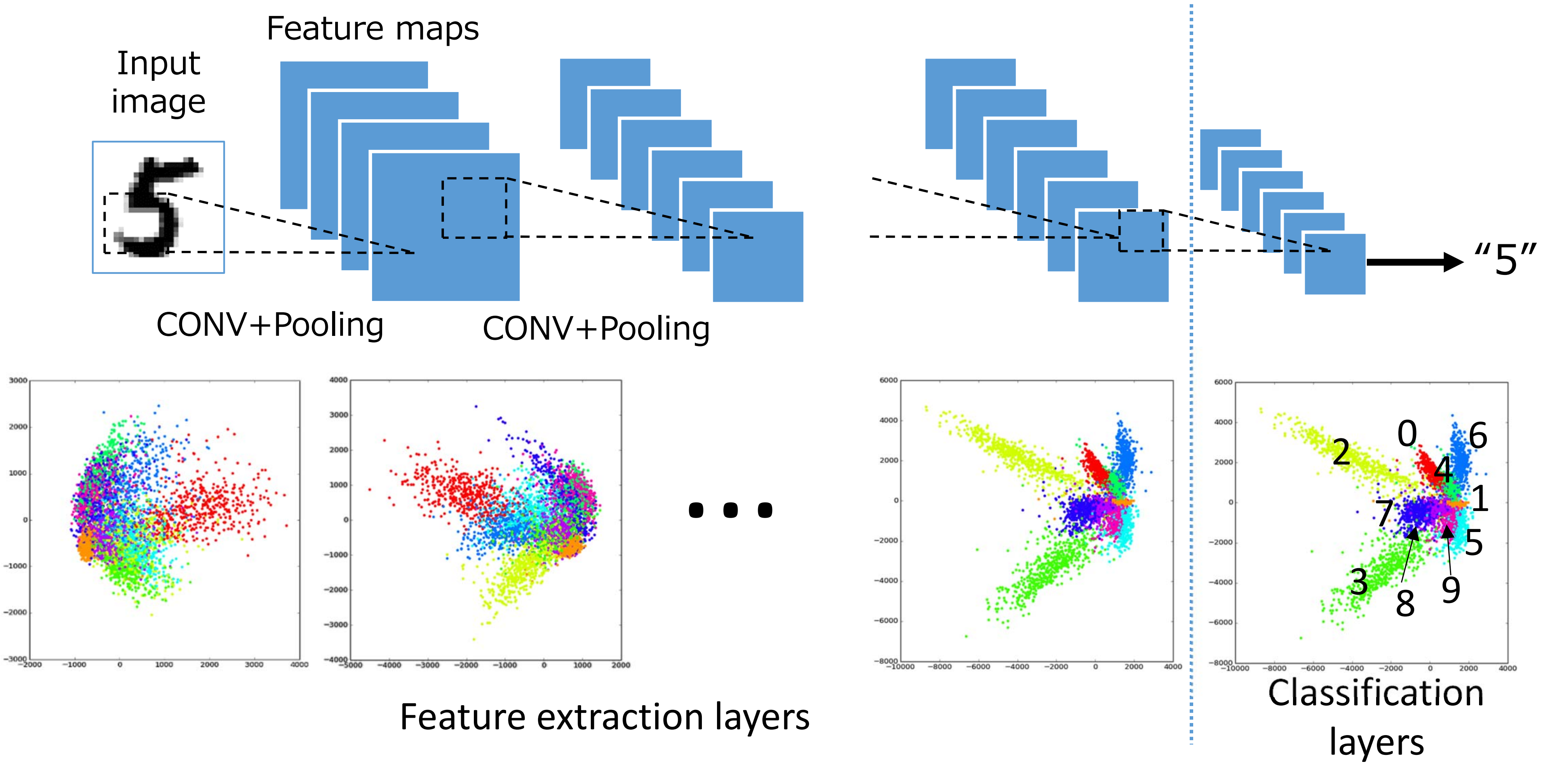




J. Dean, “Numbers everyone should know”
Source: https://gist.github.com/2841832

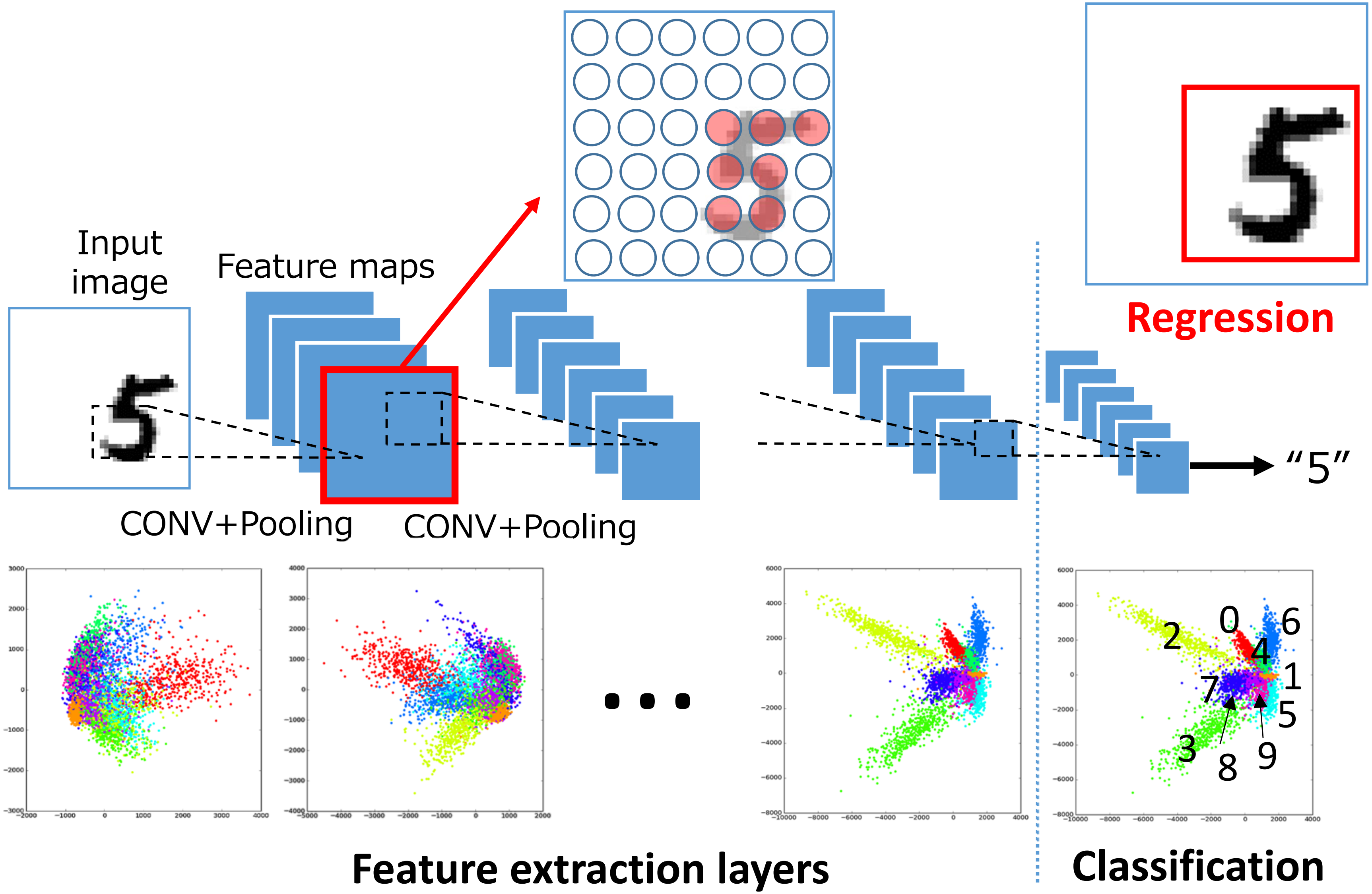
E. Joel et al., “Tutorial on Hardware Architectures for Deep Neural Networks,” MICRO-49, 2016.

Typical CNN for Classification



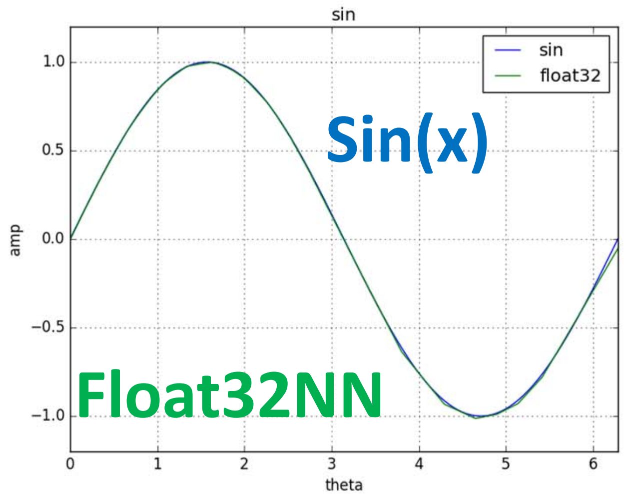
Hypothesis

- Does binarized feature map has a location? → Yes

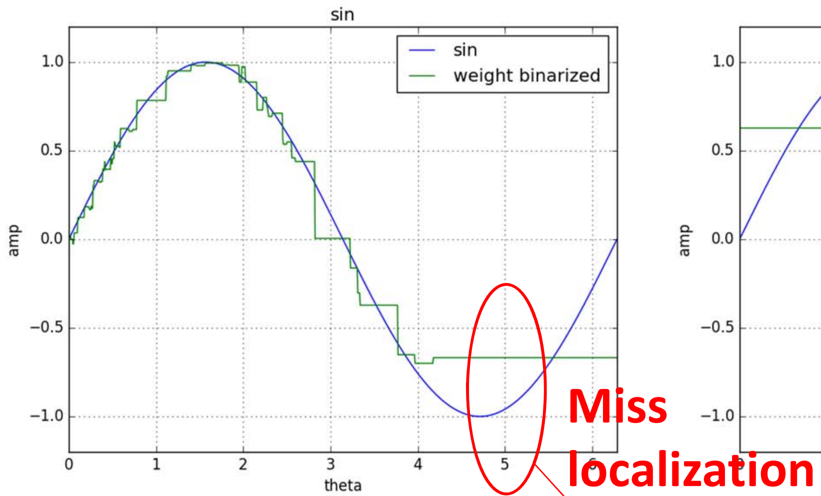


Problem

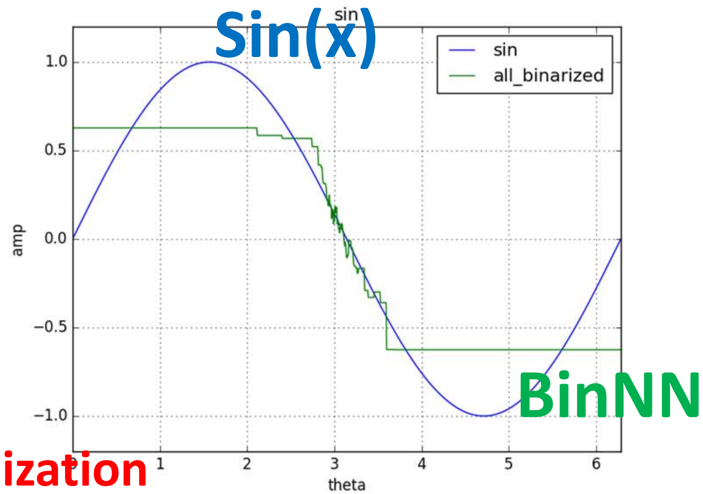
- **Low precision NN is hard to regress a function**
- **Example: sin(x) regression using a NN (3-layers)**



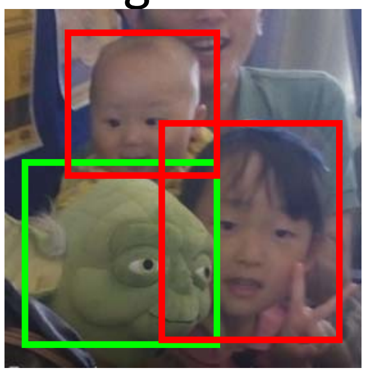
(a) Float 32 bit for activation and weight

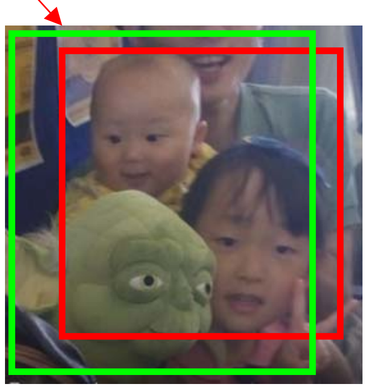


(b) Float32 for activation and binary weight



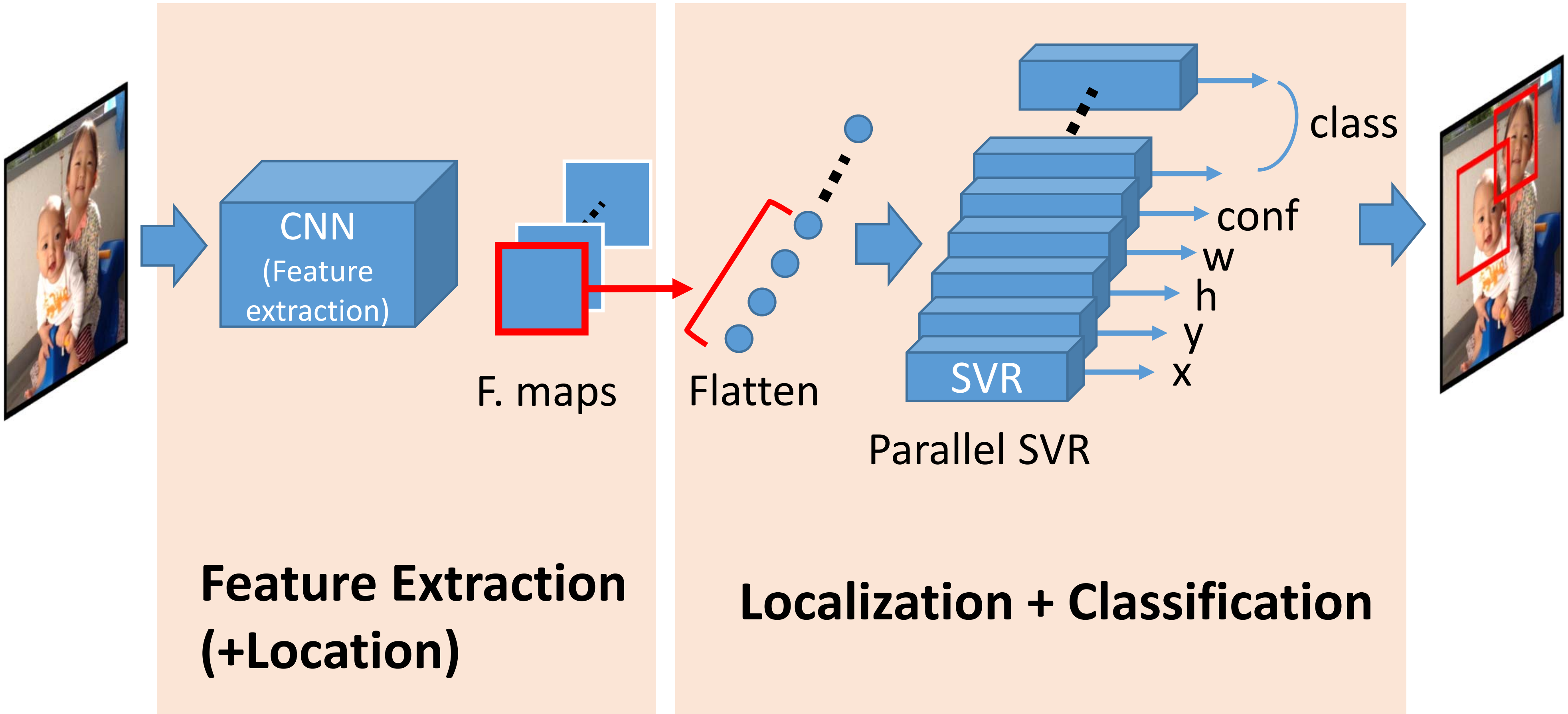
(c) All binarized





Proposed YOLOv2

- **Feature extraction layer: Binary precision**
- **<u>Localization and classification layer: Half precision</u>**



Support Vector Regression (SVR)

- Regression version of the Support Vector Machine (SVM)*1
- Passive aggressive (On-line) training is supported*2
- Model decompression (sparse like) can be applied*3

$$y = \sum_{i=1}^{n} \langle w, x_i \rangle + b$$

w : weight, x_i : i-th input, and b: bias.

*1 H. Drucker, C. J. C. Burges, L. Kaufman, A. J. Smola and V. N. Vapnik, “Support Vector Regression Machines," *Neural Information Processing Systems*, No. 9, *NIPS* 1996, pp. 155-161, 1997.

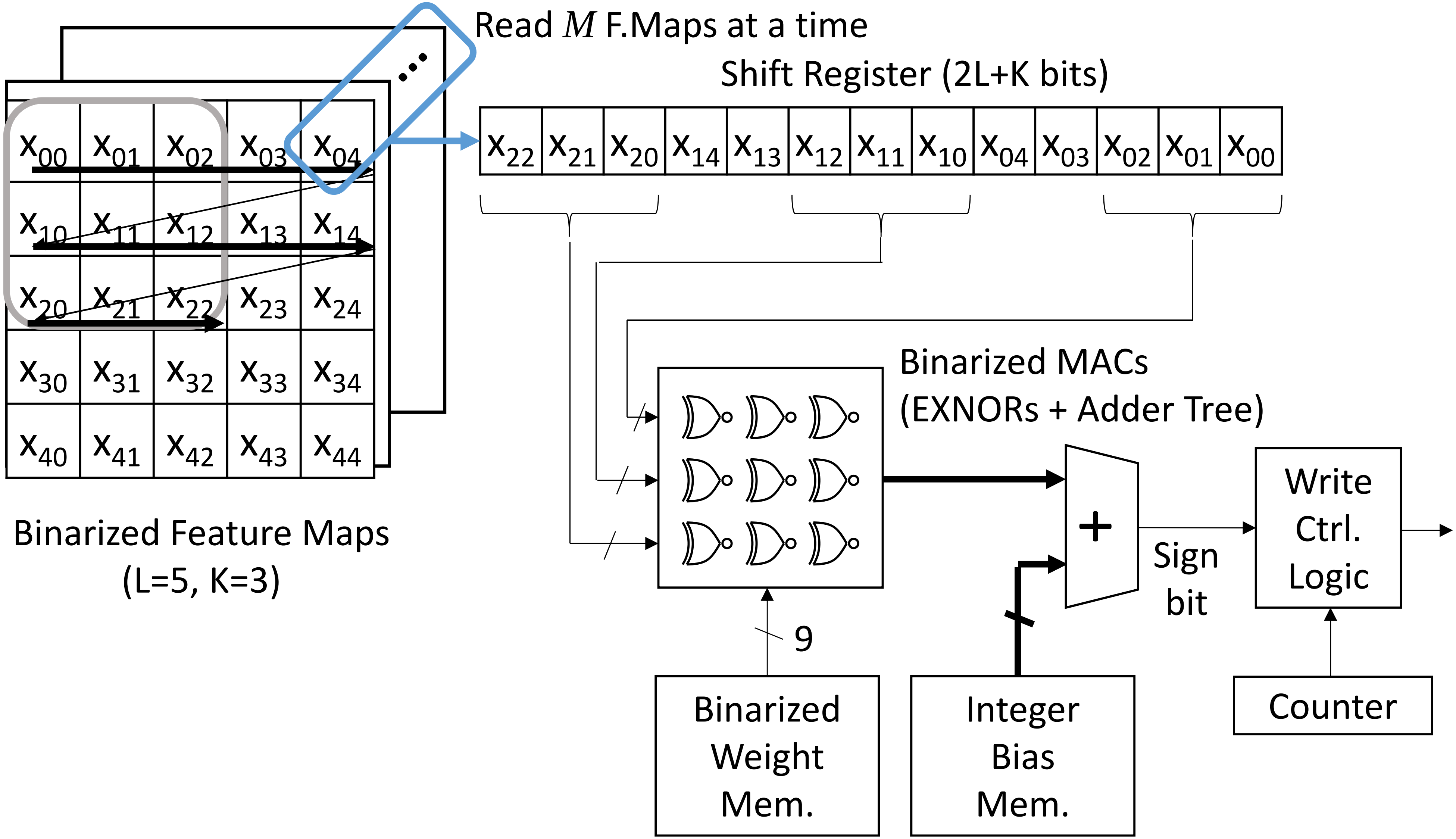
*2 M. Martin, “On-Line Support Vector Machine Regression," *ECML*, pp.282-294, 2002.

*3 T. Downs, K. E. Gates and A. Masters, “Exact simplication of support vector solutions," *Journal of Machine Learning Research*, Vol. 2, 2001, pp. 293-297.

Outline

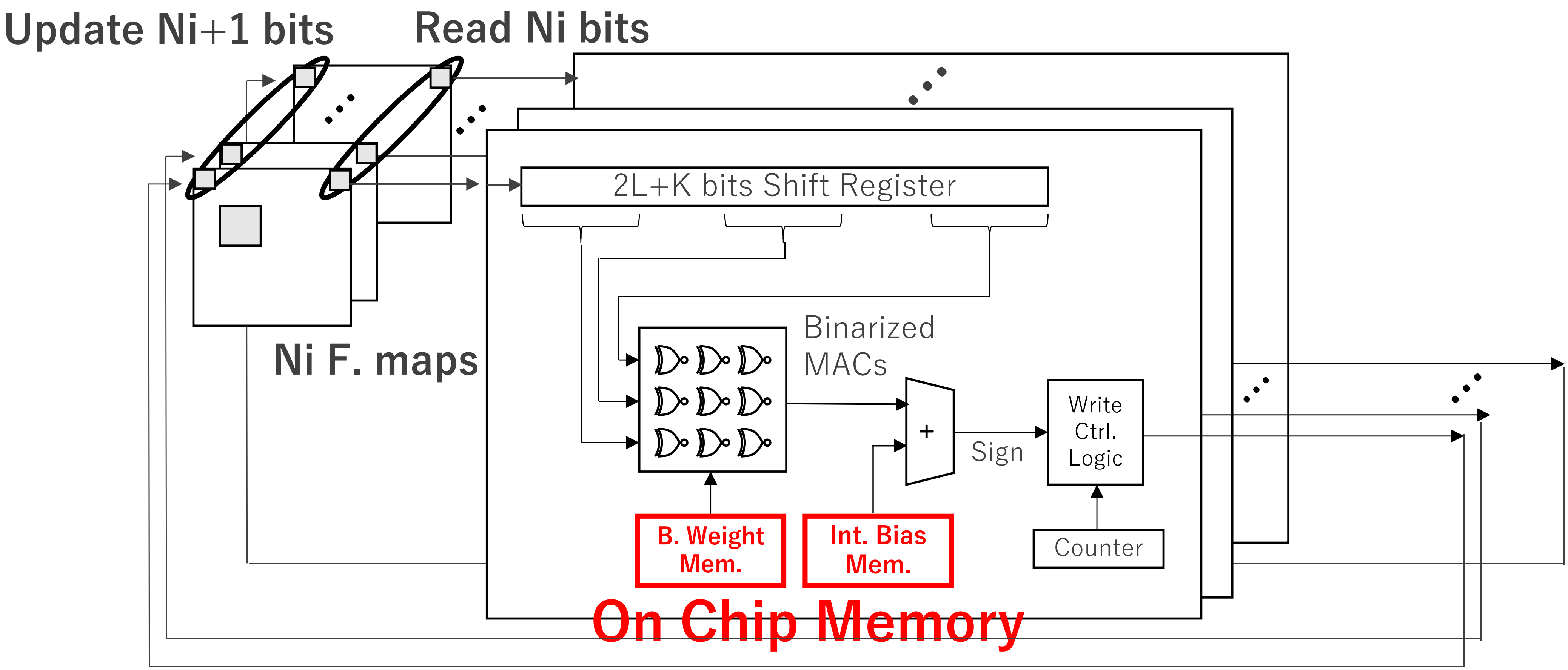
- Background
  - Convolutional Neural Network (CNN)
- Mixed-precision CNN for a Lightweight YOLOv2
  - Binary precision CNN
  - Half precision support vector regression (SVR)
- FPGA Implementation
- Experimental Results
- Conclusion

Pipelined Conv2D Circuit

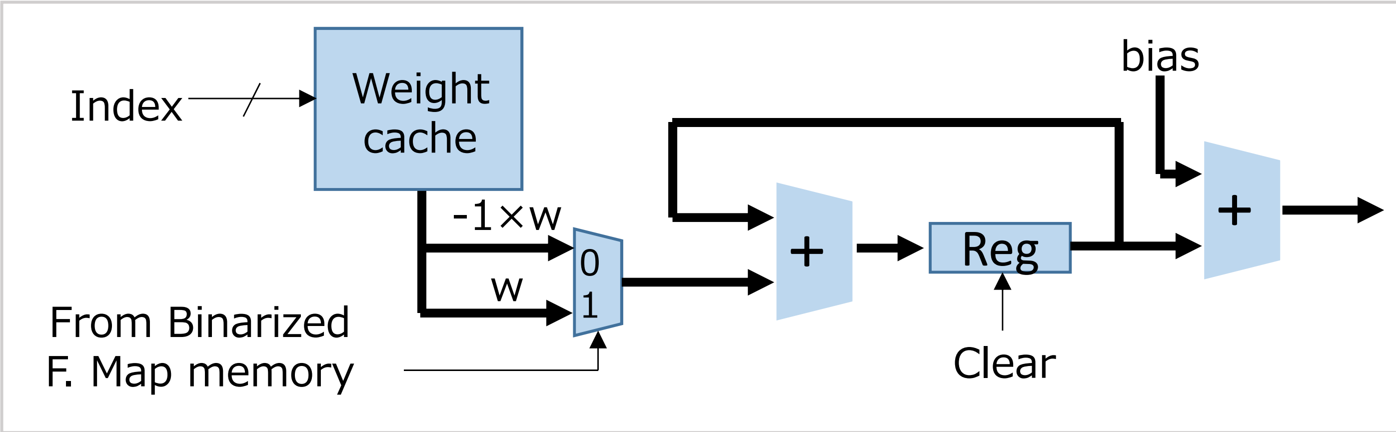


B. Bosi, G. Bois and Y. Savaria, "Reconfigurable pipelined 2-D convolvers for fast digital signal processing," *IEEE Trans. on Very Large Scale Integration (VLSI) Systems*, Vol. 7, No. 3, pp. 299-308, 1999.

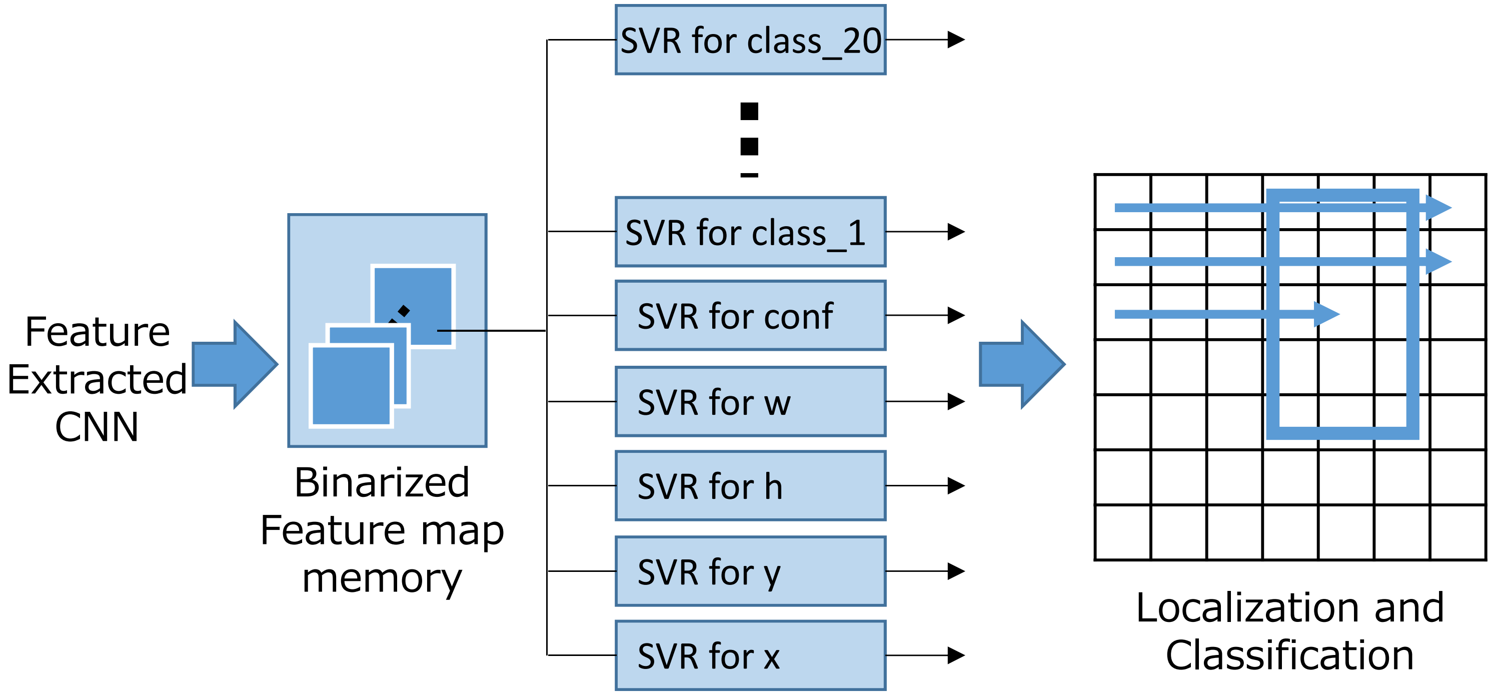
Parallel Binarized CNN Circuit



Parallel SVR Circuit

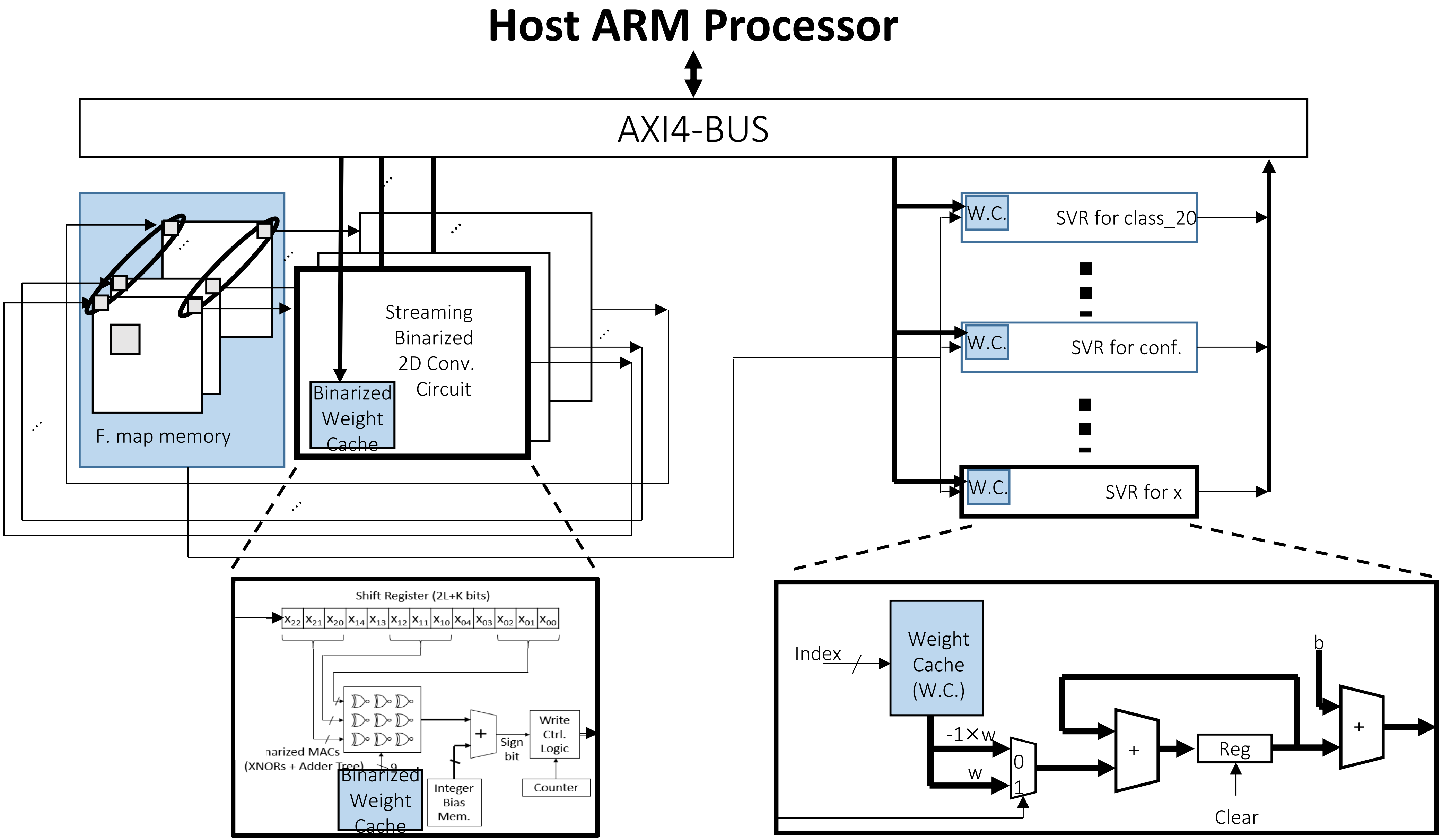


Circuit for a SVR



Parallel SVR

Overall Architecture

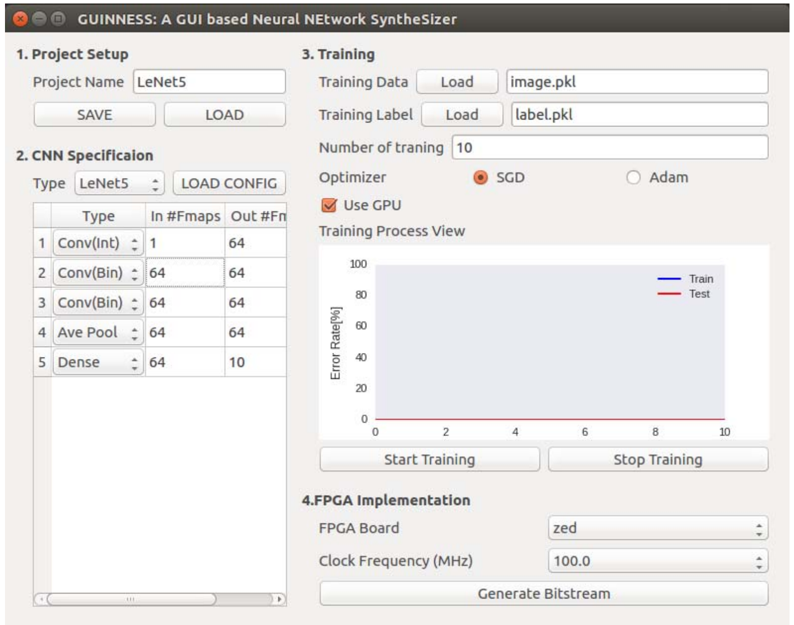


Outline

- Background
    - Convolutional Neural Network (CNN)
- Mixed-precision CNN for a Lightweight YOLOv2
    - Binary precision CNN
    - Half precision support vector regression (SVR)
- FPGA Implementation
- Experimental Results
- Conclusion

Training Result

- Environment
  - CNN: Proposed YOLOv2
    - Binarized Darknet19+SVR
  - Dataset: Pascal VOC2007
    - 21 class, 224x224 image size
- Framework
  - Binary precision CNN: GUINNES*1
  - Half precision SVR: Pegasos*2
- Accuracy (mAP)
  - 67.6%



*1 H. Nakahara et. al, "A demonstration of the GUINNESS: A GUI based neural NEtwork SyntheSizer for an FPGA," *FPL*, 2017, page 1. https://github.com/HirokiNakahara/GUINNESS
*2 S. S. Shwartz et. al, "Pegasos: primal estimated sub-gradient solver for SVM," Mathematical Programming, Vol . 127, No. 1, 2011, pp. 3-30. https://github.com/ejlb/pegasos

Implementation Setup

- Board: Xilinx Inc. Zynq UltraScale+ MPSoC zcu102 evaluation board
  - Zynq UltraScale+ MPSoC FPGA
- Design tool: SDSoC 2017.4
  - Timing constraint: 299.97MHz

Module	#LUTs	#FFs	#18Kb BRAMs	#DSP48Es
Binary CNN (2D bin. Conv.)	108,138 (103,924)	358,868 (313,839)	1680 (0)	135 (0)
Parallel SVR	27,243	11,431	26	242
Total (%)	135,381 (49.3)	370,299 (67.5)	1,706 (93.5)	377 (14.9)

Comparison

NVidia Jetson TX2

Xilinx ZCU102

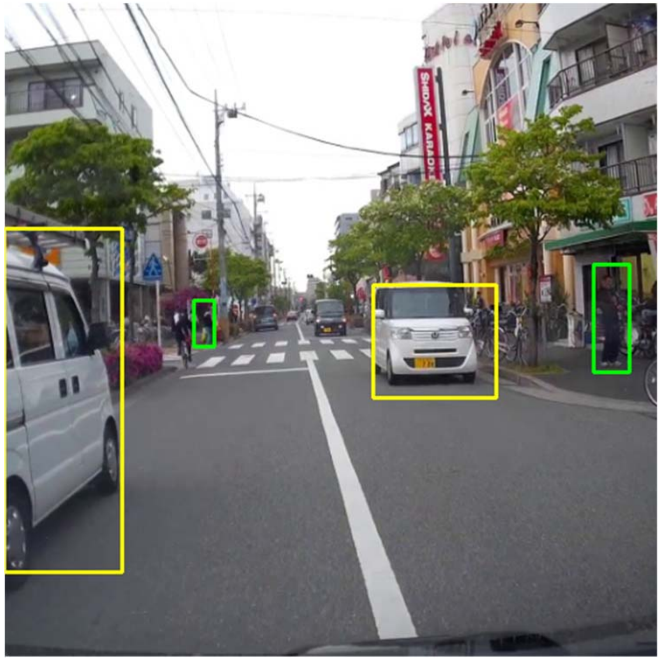
Platform	Embedded CPU	Embedded GPU	FPGA
Device	Quad-core ARM Cortex-A57	256-core Pascal GPU	Zynq UltraScale+ MPSoC
Clock Freq. [GHz]	1.9	1.3	0.3
Memory	32 GB eMMC Flash	8GB LPDDR4	32.1 Mb BRAM
Time [msec] (FPS) [sec^{-1}]	4210.0 (0.23)	715.9 (1.48*)	24.5 (40.81)
Dynamic Power [W]	4.0	7.0	4.5
Efficiency [FPS/W]	0.057	0.211	9.060

* Chainer (version 1.24.0), source code: https://github.com/leetenki/YOLOv2

Conclusion

- Lightweight YOLOv2 for a real-time object detector
  - Mixed-precision CNN
    - Binary precision CNN: Feature extraction
    - Half precision SVR: Classification and localization
- FPGA Implementation
  - Outperforms an embedded GPU and a CPU
- Future Work: Applied to CNN-based applications
    - Single-shot object detector (SSD, PVANet)
    - Semantic segmentation (FCN, U-Net)
    - Pose estimation (OpenPose)
    - CNN SLAM

Thank you!



Contact:
Hiroki Nakahara, Tokyo Institute of Technology
nakahara@ict.e.titech.ac.jp