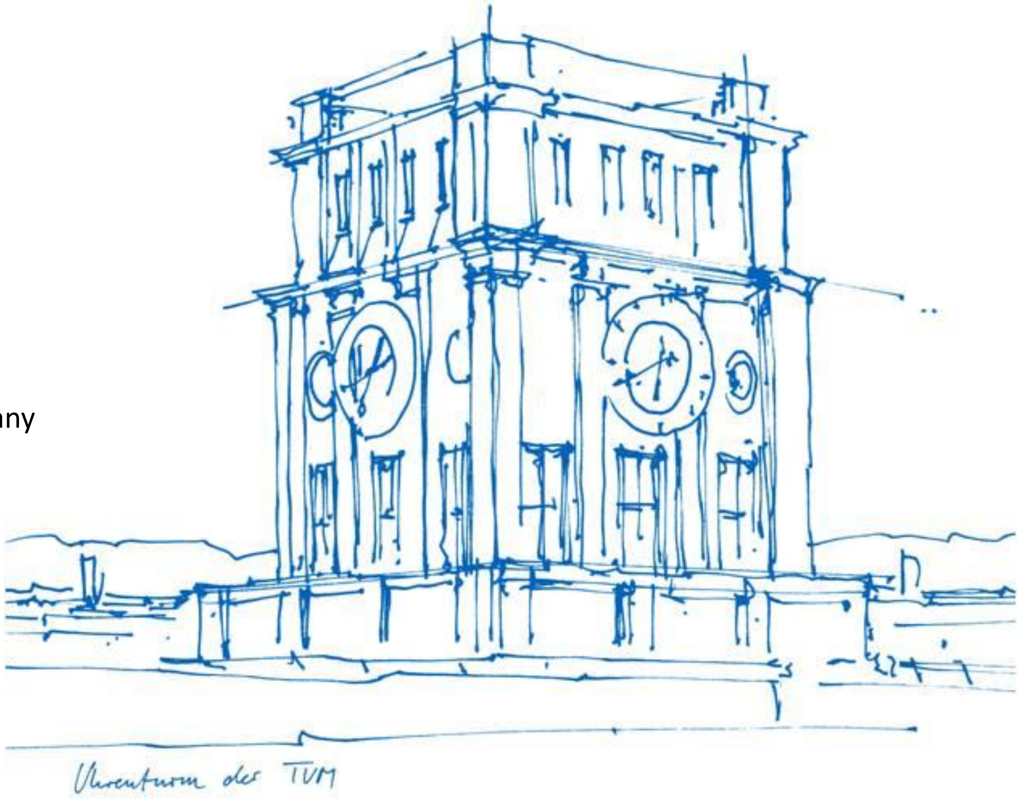
Chair of Integrated Systems
Department of Electrical and Computer Engineering
Technical University of Munich

BinaryCoP: Binary Neural Network-based COVID-19 Face-Mask Wear and Positioning Predictor

Team Number: xohw21-142
Team Members: Nael Fasfous[1], Manoj Rohit Vemparala[2], Alexander Frickenstein[2]
Supervisor: Walter Stechele[1]

[1]: Department of Electrical and Computer Engineering, Technical University of Munich, Munich, Germany
[2]: Autonomous Driving, BMW Group, Munich, Germany

[1]{nael.fasfous, walter.stechele}@tum.de
[2]{manoj-rohit.vemparala, alexander.frickenstein}@bmw.de

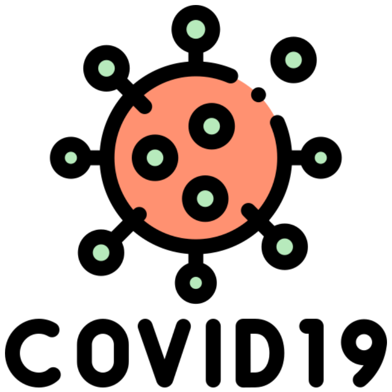


Outline

- **Problem Statement**
- **Common Challenges for AI Deployment**
  - Synthetic Data
  - Binary Neural Networks
  - Algorithm Interpretability
- **Hardware Dimensioning**
- **Comparison with Existing Work**

Problem Formulation

- Face-masks are a simple yet effective solution to mitigate the spread of COVID-19 [1]
- Most public indoor spaces have a mandatory rule on wearing masks
- **Compliance** to rules is hard to guarantee
- **Correct wearing position** of masks hard to assert



[1] When and how to use masks. [Online]. Available: https://www.who.int/emergencies/diseases/novel-coronavirus-2019/advice-for-public/when-and-how-to-use-masks

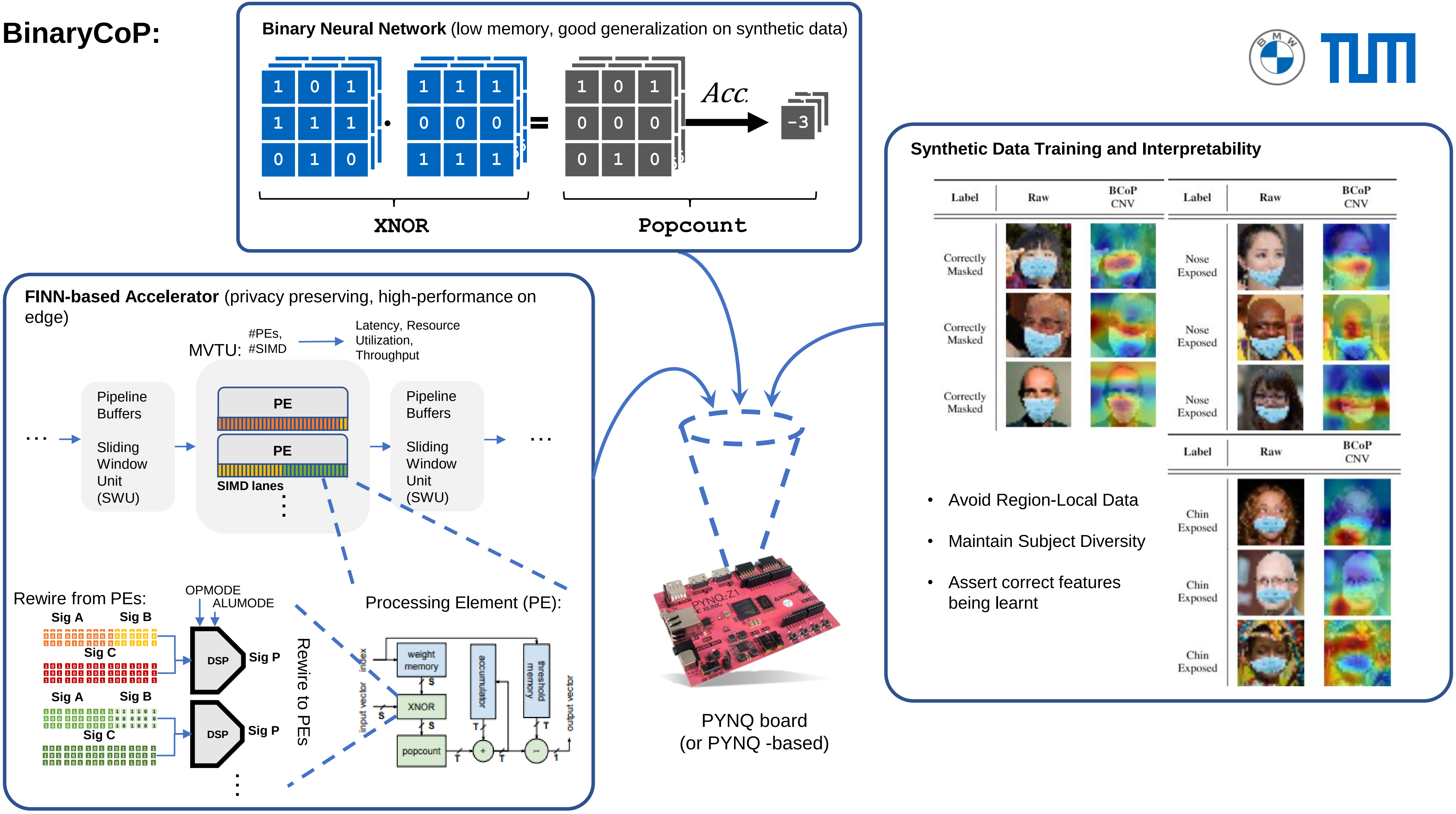
Problem Statement

Deploy **accurate, unbiased** image classification algorithms, which can be used at entrances or speed-gates to check **correct mask wear and positioning**, with all processing on **low-power, cheap, edge** hardware to **preserve privacy** of passing users.

Common Challenges for AI Deployment

Diverse datasets are hard to collect/obtain	Diversity →	Train on synthetic dataset and focus on generalizability
Low-power AI on edge requires cloud computing	Privacy/Security →	Up to 6400 FPS at 2.1W of energy on embedded FPGA
AI is unpredictable (Black box)	Explainability →	Grad-CAM for interpretability and feature analysis

BinaryCoP:



Binary Neural Network (low memory, good generalization on synthetic data)
1 0 1
1 1 1
0 1 0
1 1 1
0 0 0
1 1 1
1 0 1
0 0 0
0 1 0
Acc.
-3
XNOR
Popcount
FINN-based Accelerator (privacy preserving, high-performance on edge)
MVTU:
#PEs, #SIMD
Latency, Resource Utilization, Throughput
Pipeline Buffers
Sliding Window Unit (SWU)
PE
PE
SIMD lanes
Pipeline Buffers
Sliding Window Unit (SWU)
Rewire from PEs:
OPMODE
ALUMODE
Sig A
Sig B
Sig C
DSP
Sig P
Sig A
Sig B
Sig C
DSP
Sig P
Rewire to PEs
Processing Element (PE):
weight memory
index
input vector
XNOR
popcount
accumulator
threshold memory
output vector
PYNQ board
(or PYNQ -based)
Synthetic Data Training and Interpretability
Label
Raw
BCoP CNV
Correctly Masked
Correctly Masked
Correctly Masked
Nose Exposed
Nose Exposed
Nose Exposed
Chin Exposed
Chin Exposed
Chin Exposed
Avoid Region-Local Data
Maintain Subject Diversity
Assert correct features being learnt

Dataset Diversity

- **Diversity**
  - Large-scale **natural face** datasets exist
  - But the number of real-world **masked face images** is limited

Large dataset predominantly collected in East Asia [2]

Maintain Diversity

Synthetically generated on top of natural images [3]

[2] https://github.com/X-zhangyang/Real-World-Masked-Face-Dataset
[3] https://github.com/cabani/MaskedFace-Net

Dataset Diversity

Should we sacrifice real-world accuracy for diversity?

Synthetic data generation can be improved to close the gap [4]

BNNs have proven regularization characteristics [5]

Initialize on synthetic data, then fine-tune on real-world data

Synthetically added mask type: [4]

Mask pattern variation:

Mask color and intensity:

[4] https://github.com/aqeelanwar/MaskTheFace
[5] Courbariaux, M., Bengio, Y., David, J.P.: Binaryconnect: Training deep neural networks with binary weights during propagations

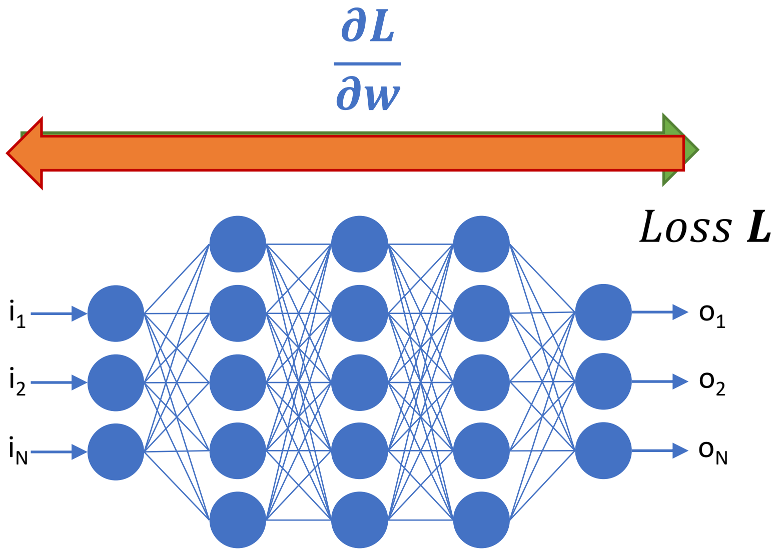
Binary Neural Networks

- Restrict all weights $\boldsymbol{w}$ and activations $\boldsymbol{a}$ to $\{-1, 1\}$ the $sign$ function:

$$b = sign(w) = \begin{cases} -1, & w < 0 \\ 1, & w \geq 0 \end{cases}$$

- Where $\boldsymbol{w}$ is the **latent weight** representation (backward-pass) and $\boldsymbol{b}$ is the **binarized weight** (forward-pass)
- **Latent weights** $\boldsymbol{w}$ are represented in **full-precision** to allow fine adjustments during training
- In final deployment, only $\boldsymbol{b}$ **binarized weights** are kept.

Binary Neural Networks: Gradient Flow Problem



- Inputs i $\in I$ result in outputs $o \in O$
- Outputs $o \in O$ result in a loss L when compared to true class of I
- Weights $w \in W$ are updated in backpropagation to minimize L
- Stochastic gradient decent (SGD) applies the chain rule to update each weight for a mini-batch input:

Update $\boldsymbol{w}$ to minimize $\boldsymbol{L}$
i.e. find minimization slope:

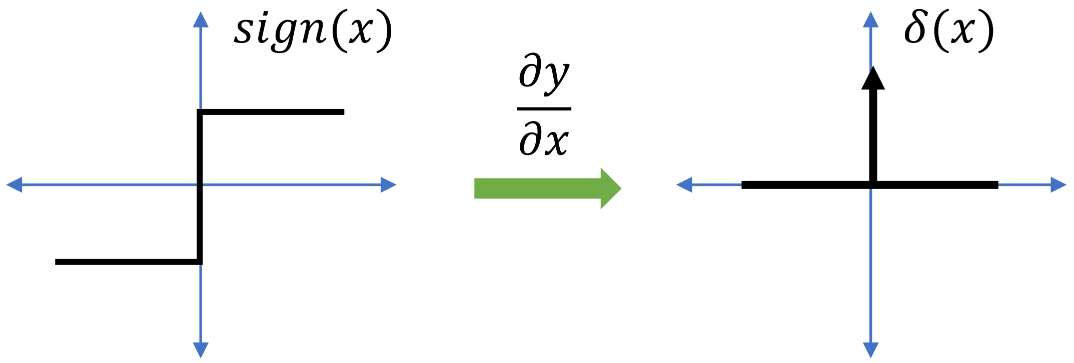
$$\frac{\partial \boldsymbol{L}}{\partial \boldsymbol{w}} = \frac{\partial v}{\partial \boldsymbol{w}} \boxed{\frac{\partial z}{\partial v} \frac{\partial k}{\partial z}} \frac{\partial \boldsymbol{L}}{\partial k}$$

Intermediate operations between $\boldsymbol{L}$ and $\boldsymbol{w}$

Binary Neural Networks

- The $sign$ function is an intermediate operation in BNNs:

$$y = sign(x) = \begin{cases} -1, & w < 0 \\ 1, & w \geq 0 \end{cases}$$



$$\frac{\partial \boldsymbol{L}}{\partial \boldsymbol{w}} = \frac{\partial v}{\partial \boldsymbol{w}} \boxed{\frac{\partial z}{\partial v} \frac{\partial k}{\partial z}} \frac{\partial \boldsymbol{L}}{\partial k}$$

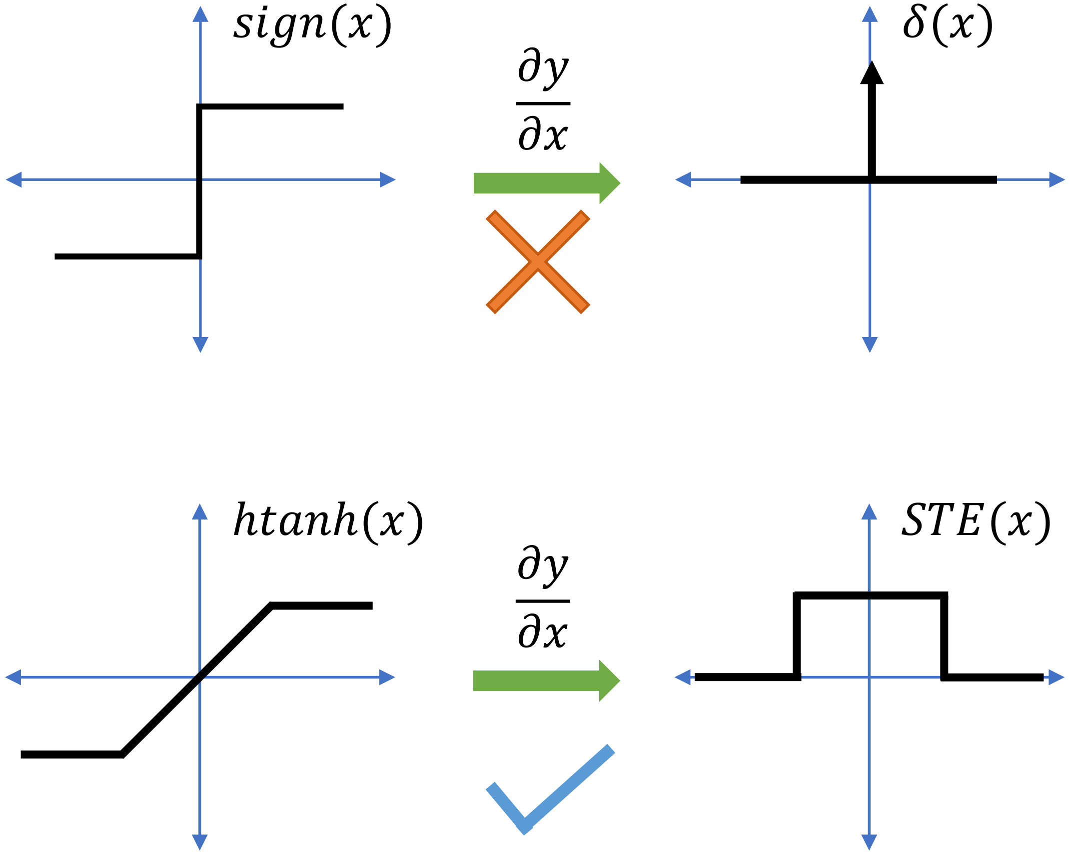
Intermediate operations between $\boldsymbol{L}$ and $\boldsymbol{w}$

Derivative of $sign$ blocks gradient flow because it is **zero almost everywhere**!

Discreteness of binarization is incompatible with **Gradient Decent optimization**

Binary Neural Networks: Gradient Flow Problem

- Solution: ***approximate*** derivative of $sign$ with $htanh$ (or other):



Gradient flow possible again!

Binary Neural Networks

- So what do we get for all this?

High-Precision		Binary
Multiplications	→	XNOR
Accumulation	→	PopCount
BatchNorm	→	Comparator
32 or 16 Bits/Operand	→	1-Bit Operands

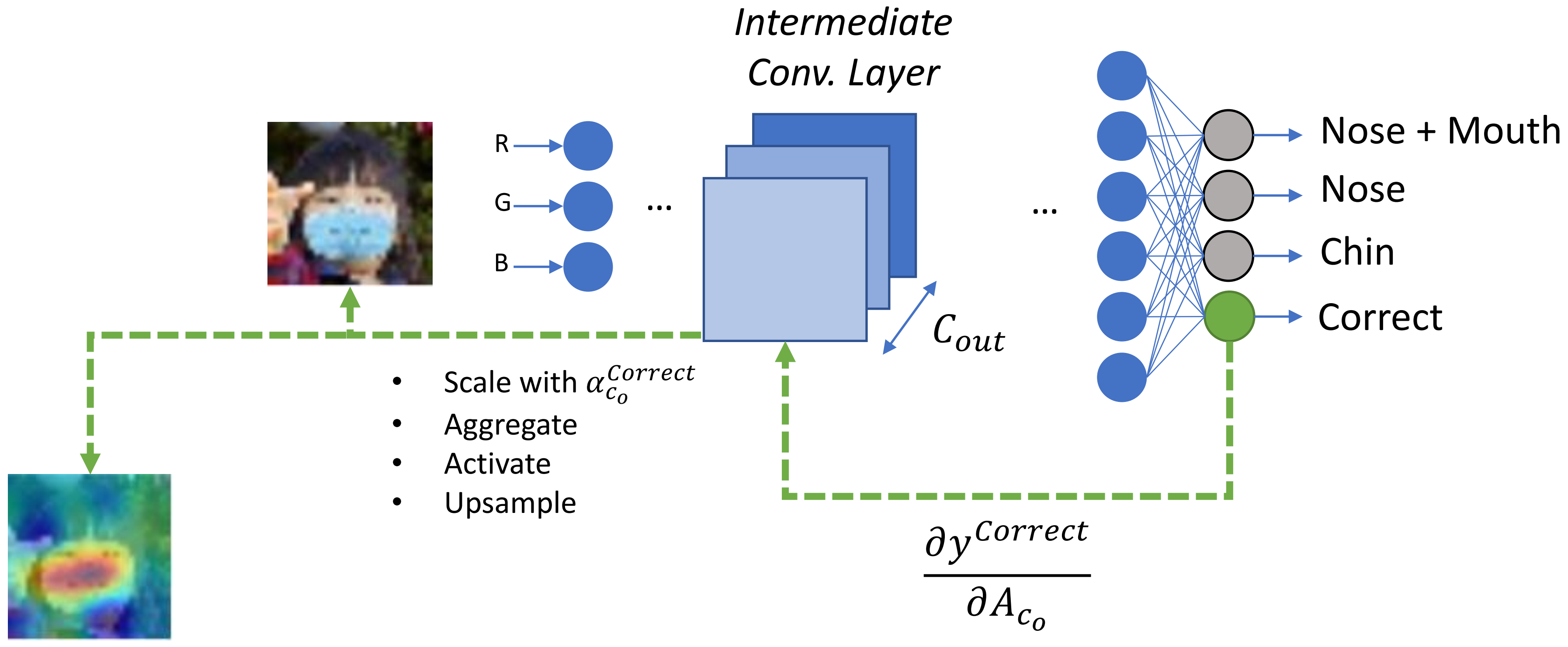
- **Bonus:**

  BNNs' approximate training makes it harder to overfit on training data

  Good candidate for training on Synthetic Data

Algorithm Explainability: Grad-CAM [6]

- Are NNs black boxes?



NN Attention

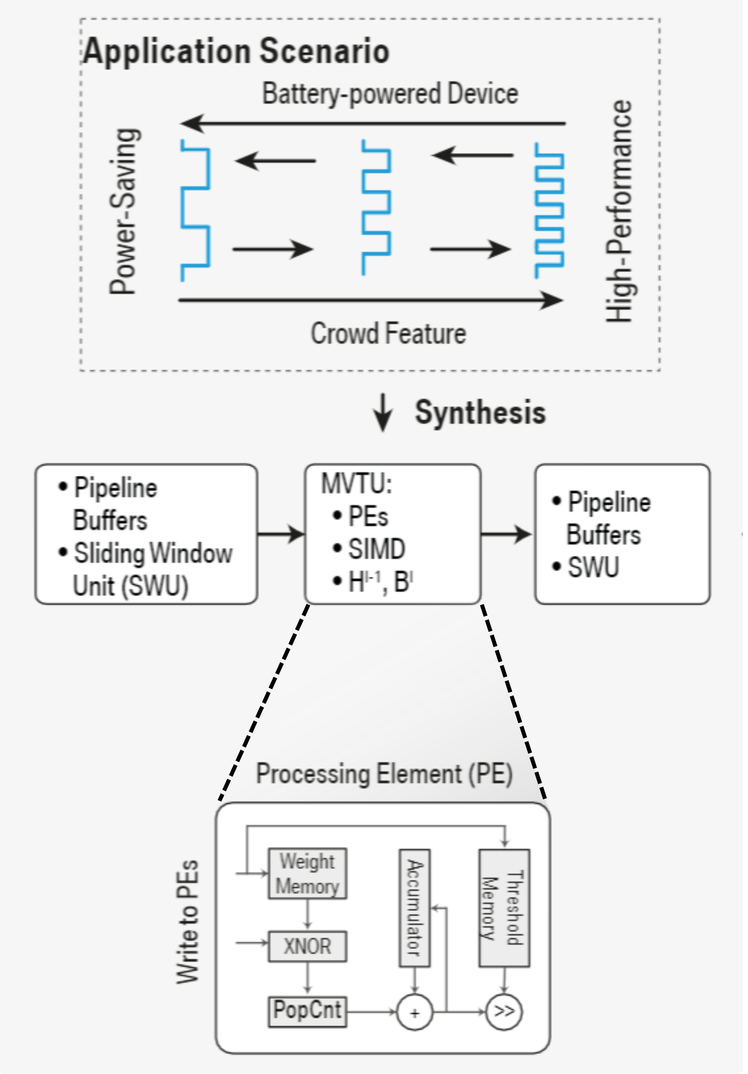
NN's learning can be observed to understand its decisions and learning patterns

$$\alpha_{c_o}^{Correct} = Glob.avg\left(\frac{\partial y^{Correct}}{\partial A_{c_o}}\right)$$ = **Neuron importance weight**

[6] R. Selvaraju et al., "Grad-CAM: Visual Explanations from Deep Networks via Gradient-based Localization"

Hardware Acceleration: FINN framework [7]

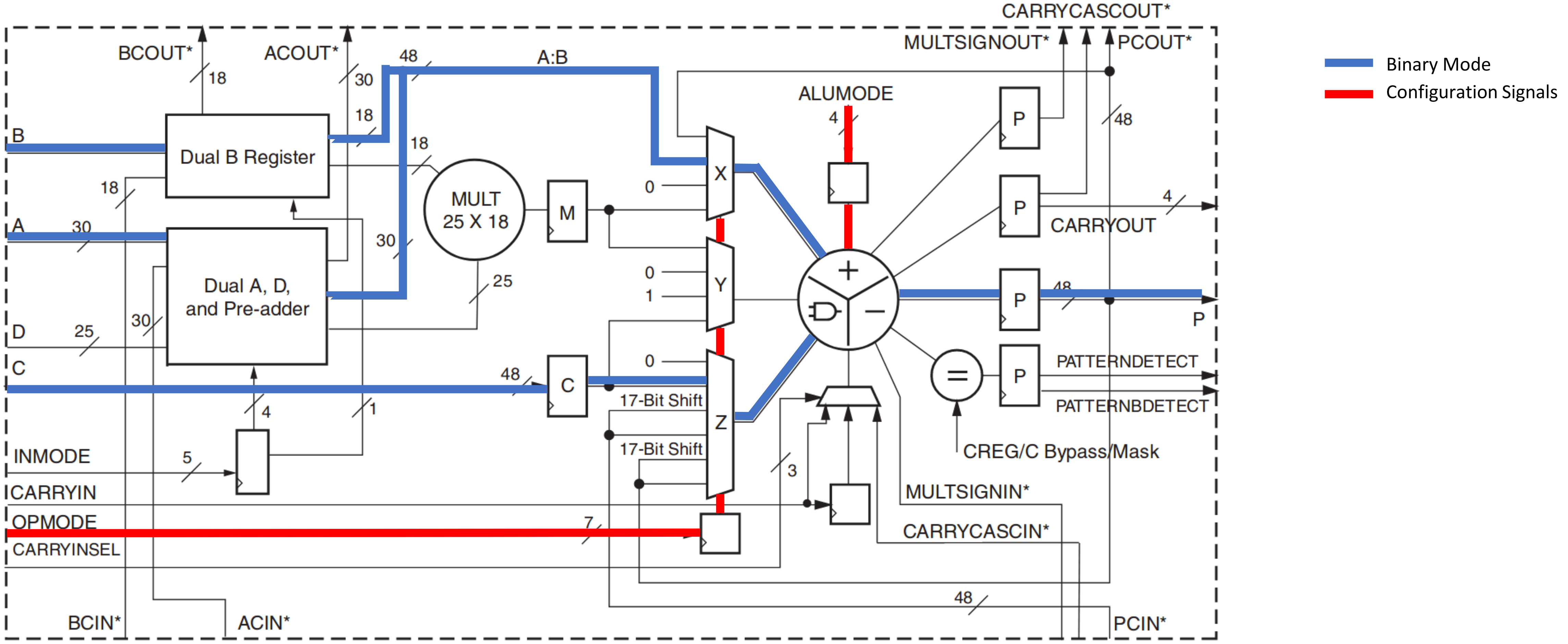
- BNNs can have a very small memory footprint
  - Accelerator can pre-load all weights
  - Computation units (MVTUs) can be parameterized (PEs, SIMD)
  - Match-throughput of MVTUs



[7] Y. Umuroglu et al., "FINN: A Framework for Fast, Scalable Binarized Neural Network Inference"

Hardware Acceleration: Bit-Packing

XNORs into DSPs



*These signals are dedicated routing paths internal to the DSP48E1 column. They are not accessible via fabric routing resources.

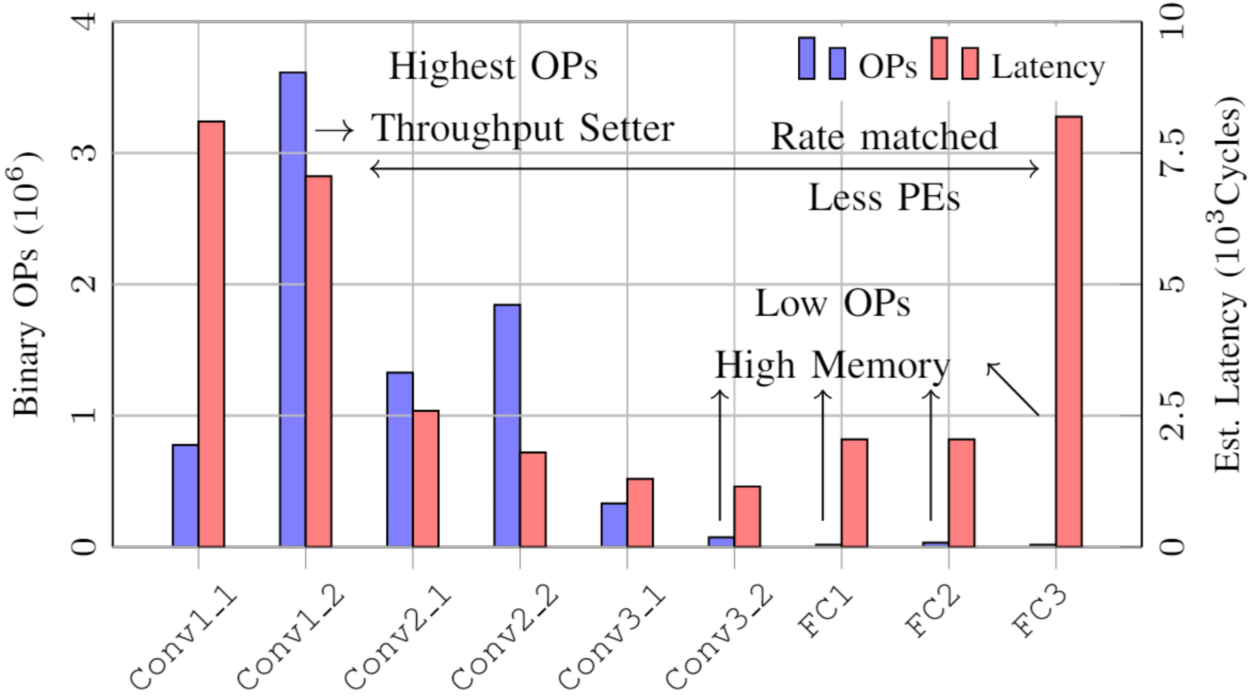
BinaryCoP Prototypes

- 3 BinaryCoP variants

	High Accuracy	Low Memory	Low Logic
Network	**CNV**	n-**CNV**	μ-**CNV**
Arch. $L \mid [C_i, C_o]$ $K = 3\ \forall$ Conv	Conv1_1 | [3, 64] Conv1_2 | [64, 64] Conv2_1 | [64, 128] Conv2_2 | [128, 128] Conv3_1 | [128, 256] Conv3_2 | [256, 256] FC1 | [512] FC2 | [512] FC3 | [4]	Conv1_1 | [3, 16] Conv1_2 | [16, 16] Conv2_1 | [16, 32] Conv2_2 | [32, 32] Conv3_1 | [32, 64] Conv3_2 | [64, 64] FC1 | [128] FC2 | [128] FC3 | [4]	Conv1_1 | [3,16] Conv1_2 | [16, 16] Conv2_1 | [16, 32] Conv2_2 | [32, 32] Conv3_1 | [32, 64] FC1 | [128] FC2 | [4]
PE Count **SIMD lanes**	16, 32, 16, 16, 4, 1, 1, 1, 4 3, 32, 32, 32, 32, 32, 4, 8, 1	16, 16, 16, 16, 4, 1, 1, 1, 1 3, 16, 16, 32, 32, 32, 4, 8, 1	4, 4, 4, 4, 1, 1, 1 3, 16, 16, 32, 32, 16, 1

BinaryCoP Prototypes

- Throughput Matching – Weight Memory Fragmentation
  - Example: BinaryCoP-*n*-CNV



Prototype Hardware Results

- Masked-FaceNet Dataset
- 32x32 input resolution

Low Power

Prototype	LUT	BRAM	DSP	Power [W] Idle	Power [W] Inf.	Thr.put [FPS]	Latency [ms]	Acc. [%]
CNV	26060	124	24		2.212	3049	1.58	**98.10**
n-CNV	20425	**10.5**	14	1.65*	2.122	**6460**	0.31	93.94
μ-CNV	**11738**	14	27		2.028	1646	0.81	93.78

*Required by the board and ARM-Cortex A9 processor. Accelerator is idle.

High Accuracy

Low Logic
(Synthesizable on Z7010)

Low Memory
High Throughput

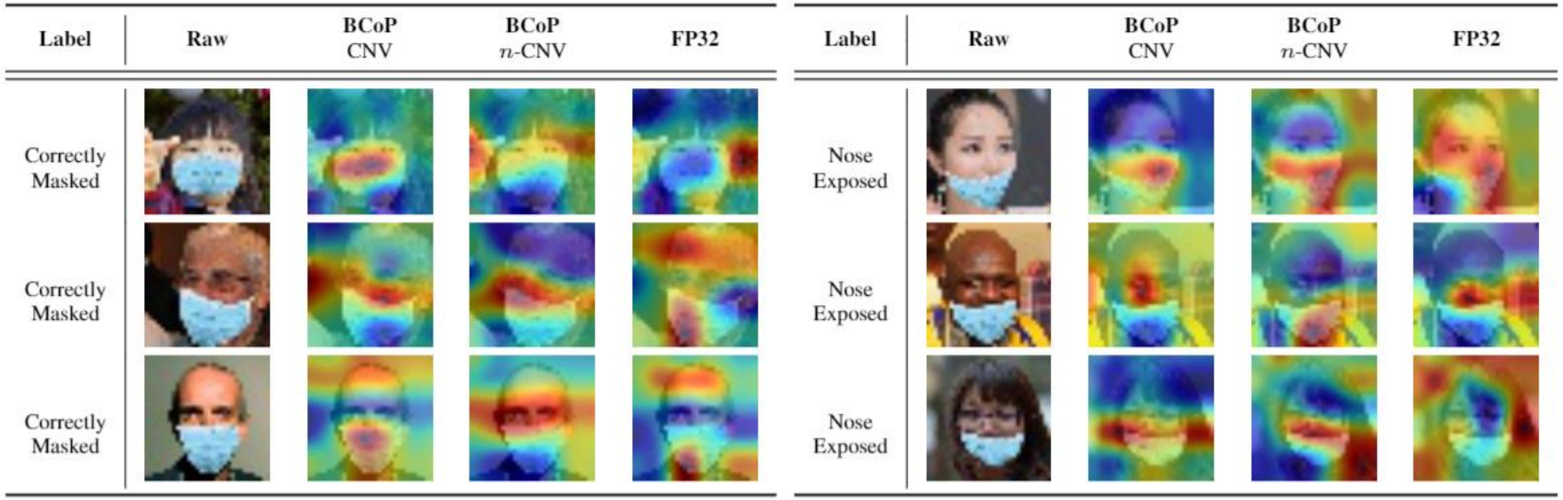
Explainability Results

- Confusion matrix of BinaryCoP-CNV

True Class \ Predicted Class	Correct	Nose	N+M	Chin
Correct	7125 98%	41 1%	1 0%	90 1%
Nose	26 0%	7042 98%	94 2%	26 0%
N+M	4 0%	79 1%	5651 98%	9 0%
Chin	107 1%	41 1%	7 0%	7363 98%

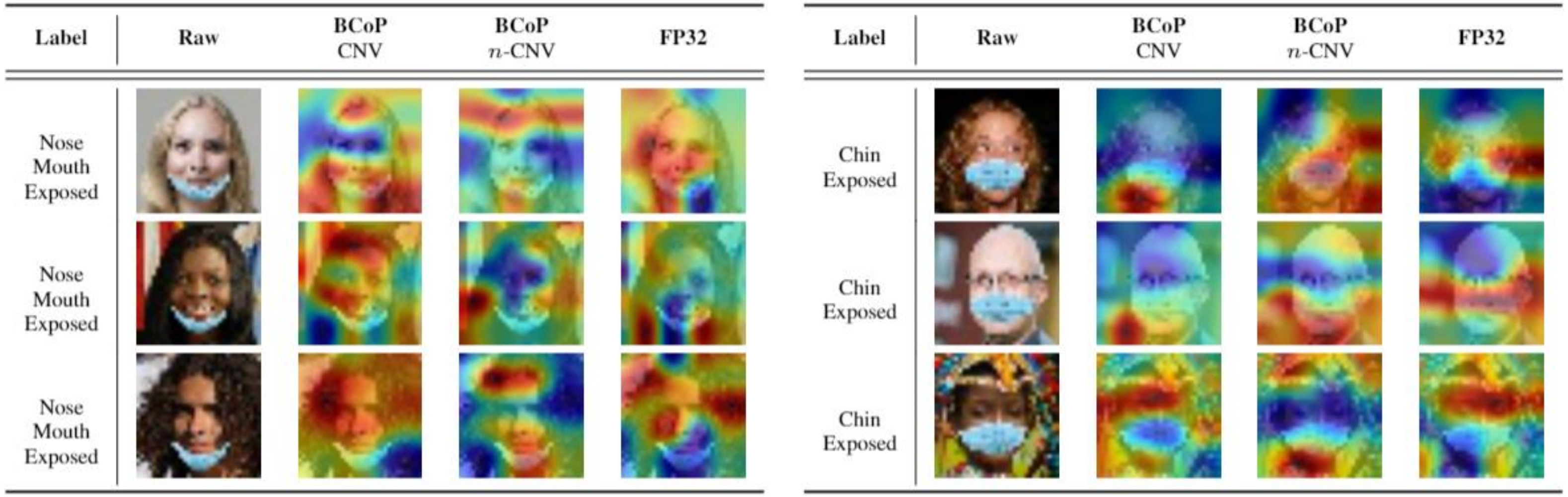
Chin area too small,
can be overlooked by BNN

Explainability Results: Grad-CAM Interpretation



(a) Correctly masked Grad-CAM

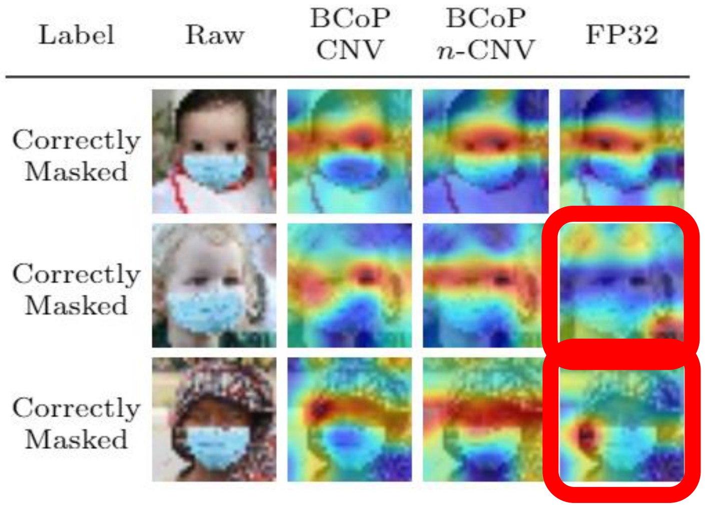
(b) Nose exposed Grad-CAM



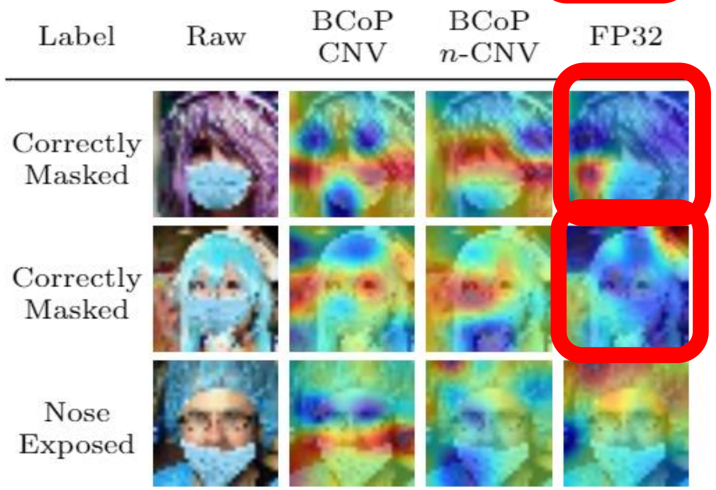
(c) Mouth + nose exposed Grad-CAM

(d) Chin exposed Grad-CAM

Explainability Results: Diversity and Generalizability

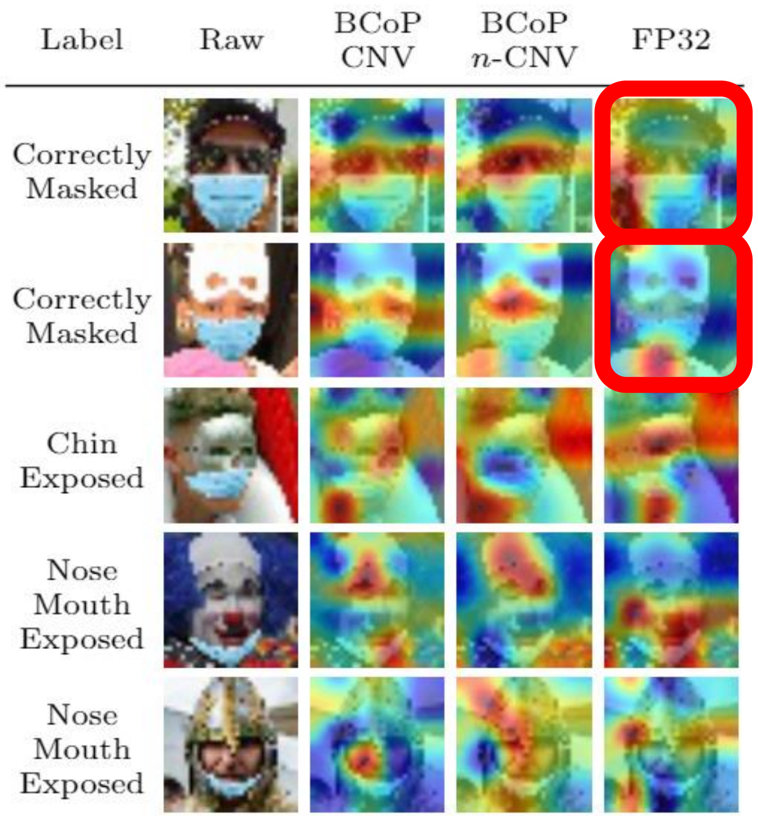


Age Generalization

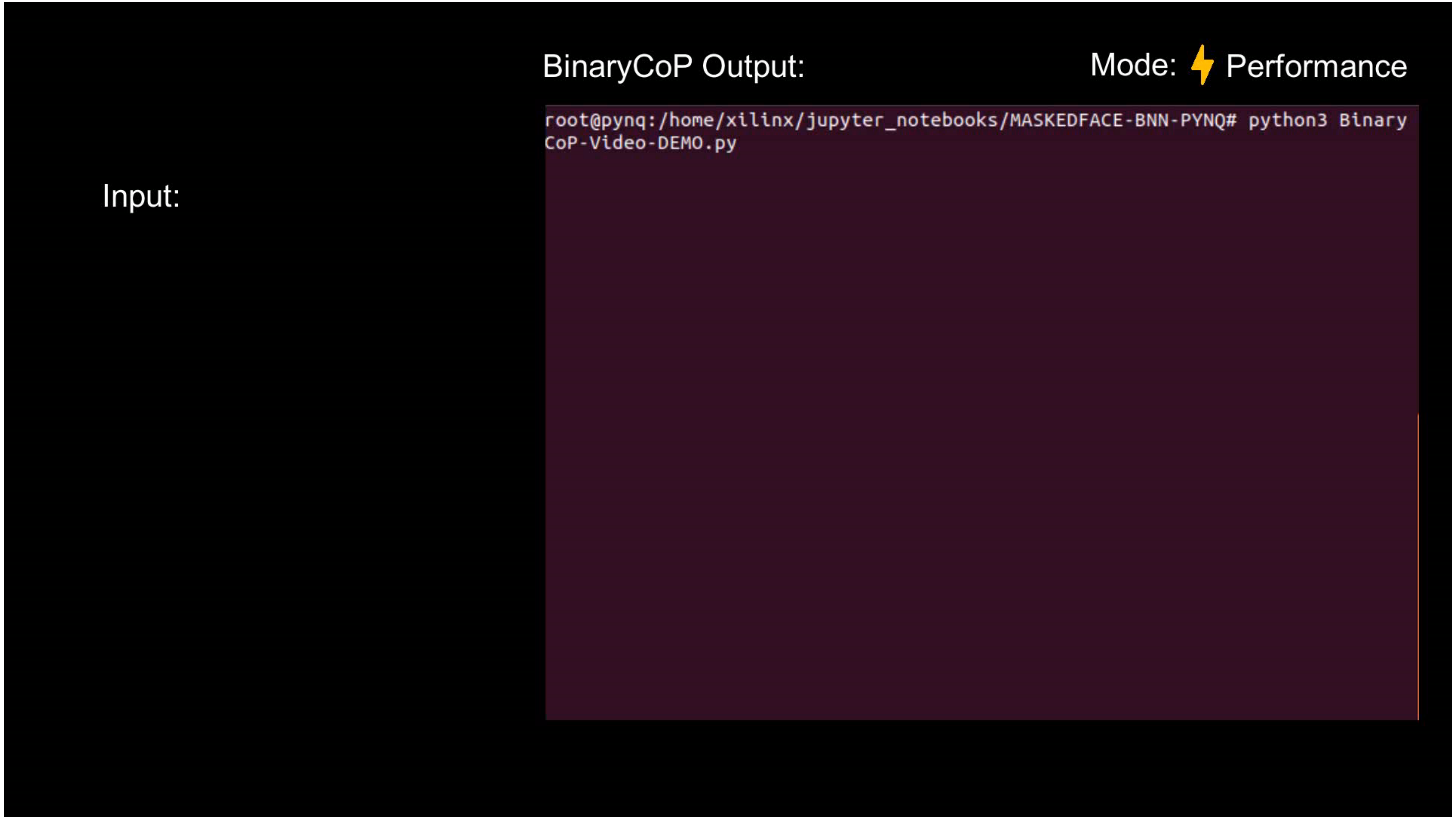


Hair/Head gear Generalization

Manipulated Face Generalization



Demo


BinaryCoP Output:
Mode: Performance
Input:
root@pynq:/home/xilinx/jupyter_notebooks/MASKEDFACE-BNN-PYNQ# python3 BinaryCoP-Video-DEMO.py

Comparison with Other Works

- Dataset for Mask Wear problem are not standardized
- Tasks range from classification to object detection
- Resolutions range from 960×544 to 32x32

Work	Pro	Con
NVIDIA [8]	• High Res • Object Localization	• 500€ Hardware • 20-30 Watts
Amazon [9]	• High Res • Other PPE supported	• Cloud Processing
Wang et al. [10]	• Serverless, In-browser Processing	• Requires Web-assembly Browser (tested on iPhone, iPad, MacBook) • $$$ Expensive Devices
"CheckYourMask" Hammoudi et al. [11]	• Edge Processing through Android app	• Designed for self-check, not surveillance • Not optimized for power, low-battery application

Conclusion

Diverse datasets are hard to collect/obtain	Diversity →	Train on synthetic dataset and focus on generalizability
Low-power AI on edge requires cloud computing	Privacy/Security →	Up to 6400 FPS at 2.1W of energy on embedded FPGA
AI is unpredictable (Black box)	Explainability →	Grad-CAM for interpretability and feature analysis

Resources and References

[1] When and how to use masks. [Online]. Available: https://www.who.int/emergencies/diseases/novel-coronavirus-2019/advice-for-public/when-and-how-to-use-masks

[2] https://github.com/X-zhangyang/Real-World-Masked-Face-Dataset

[3] https://github.com/cabani/MaskedFace-Net

[4] https://github.com/aqeelanwar/MaskTheFace

[5] M. Courbariaux et al. “Binaryconnect: Training deep neural networks with binary weights during propagations”

[6] R. Selvaraju et al., “Grad-CAM: Visual Explanations from Deep Networks via Gradient-based Localization”

[7] Y. Umuroglu et al., “FINN: A Framework for Fast, Scalable Binarized Neural Network Inference”

[8] A. Kulkarni et al., “Implementing a real-time, ai-based, face mask detector application for covid-19”

[9] T. Agrawal et al., ““Automatically detecting personal protective equipment on persons in images using amazon rekognition”

[10] Z. Wang et al, “Wearmask:Fast in-browser face mask detection with serverless edge computing for covid-19”

[11] K. Hammoudi et al., “Validating the correct wearing of protection mask by taking a selfie: Design of a mobile application “checkyourmask” to limit the spread of covid-19”

Slide 3 has been designed using resources from Flaticon.com