Improving the Energy Efficiency and Robustness of tinyML Computer Vision using Log-Gradient Input Images

Authors

Qianyun Lu
Stanford University
Stanford, United States
savylu@stanford.edu

Boris Murmann
Stanford University
Stanford, United States
murmann@stanford.edu

Presenters

Team M&Ms

Masa Cirkovic

Mete Harun Akcay

7 May, 2025

AGENDA

Problem Statement

Log∇ (Pipeline, Computation, Intuition)

Datasets

Experiments

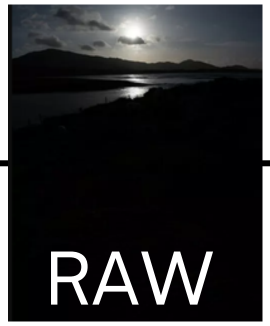
Architecture Search, Fixed Architectures

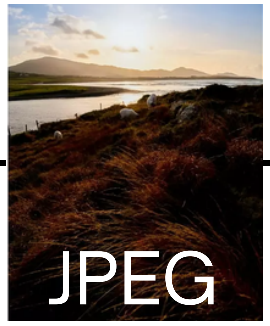
Conclusion

Problem Statement

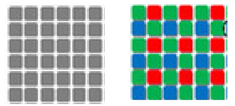
Conventional Computer Vision

Image Sensor → RAW → Image Signal Processor (ISP) → JPEG → CNN





- Monochrome
- Bayer



- Demosaicking
- Gama correction
- White balancing

- Segmentation
- Object detection
- Classification

- designed for human perception
- more costly
- more energy consumption
- longer processing time

inefficient.

TinyML Computer Vision

“Why not feed the neural network only what it needs — and cut out the rest?”

idea: improve each component

Image Sensor

- Smaller sensors
- Lower-power sensors

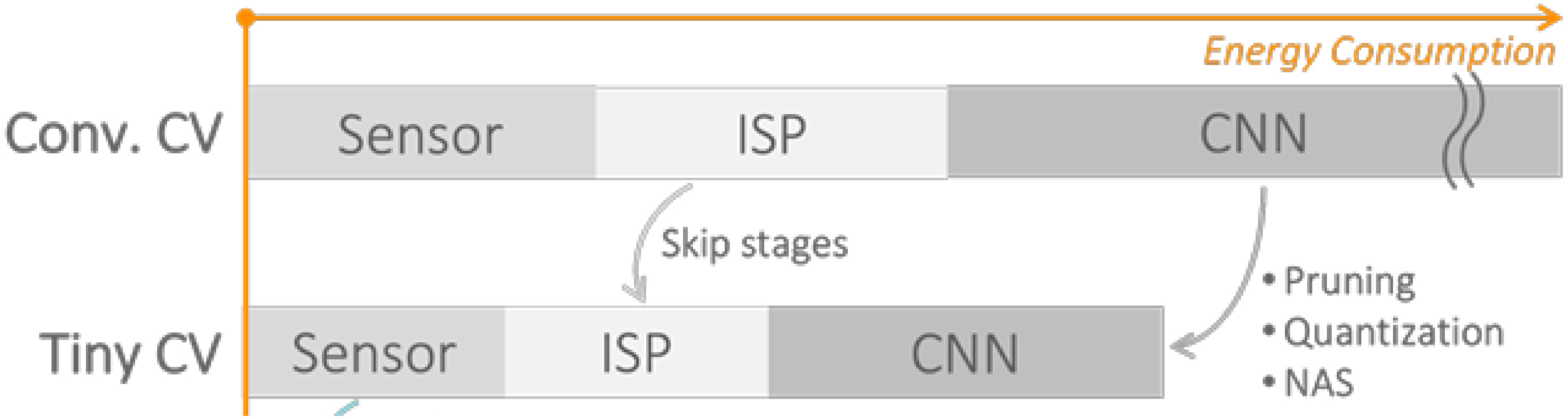
ISP

- Skip stages

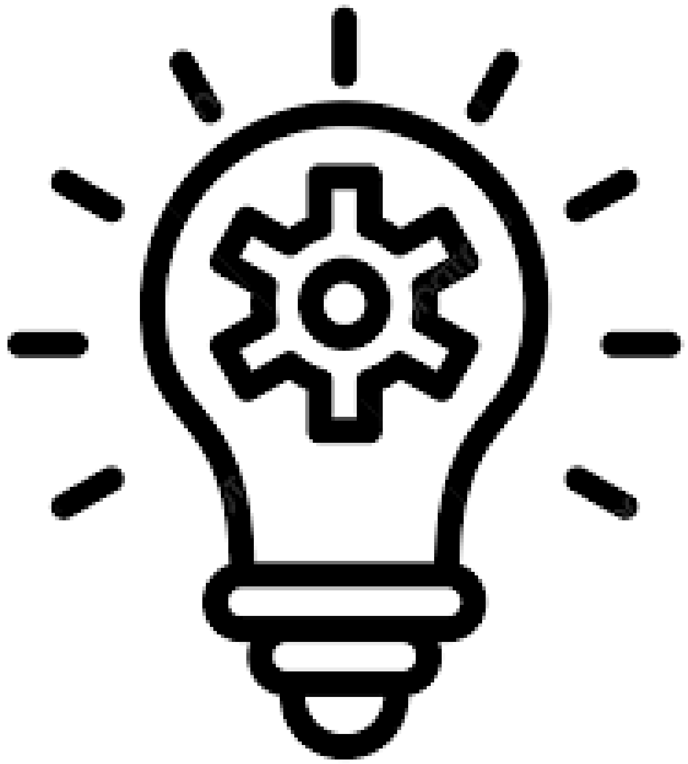
CNN

- Pruning
- Quantization
- NAS

Problem Statement

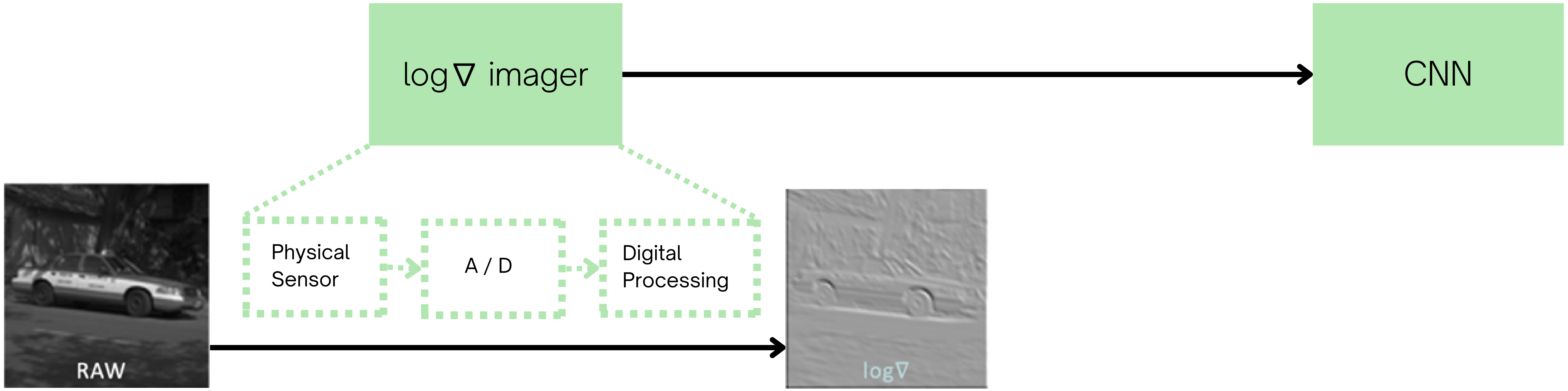


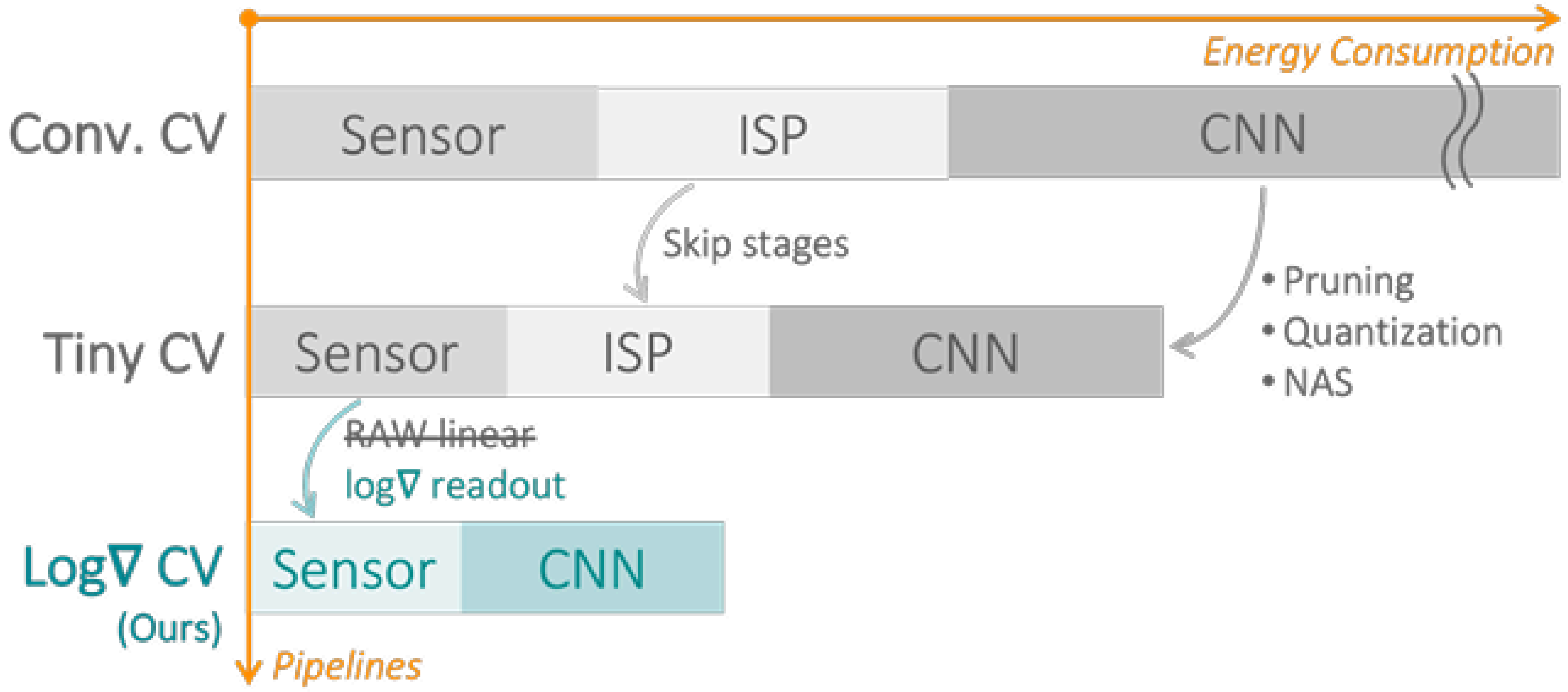
- improvements are isolated
- still not enough efficiency



Feed log-gradient images directly to the CNN — and skip everything else!

Solution: **log∇**





- no ISP layer
- more robust
- more efficient
- more CNN-compressibility

Log∇ Computation

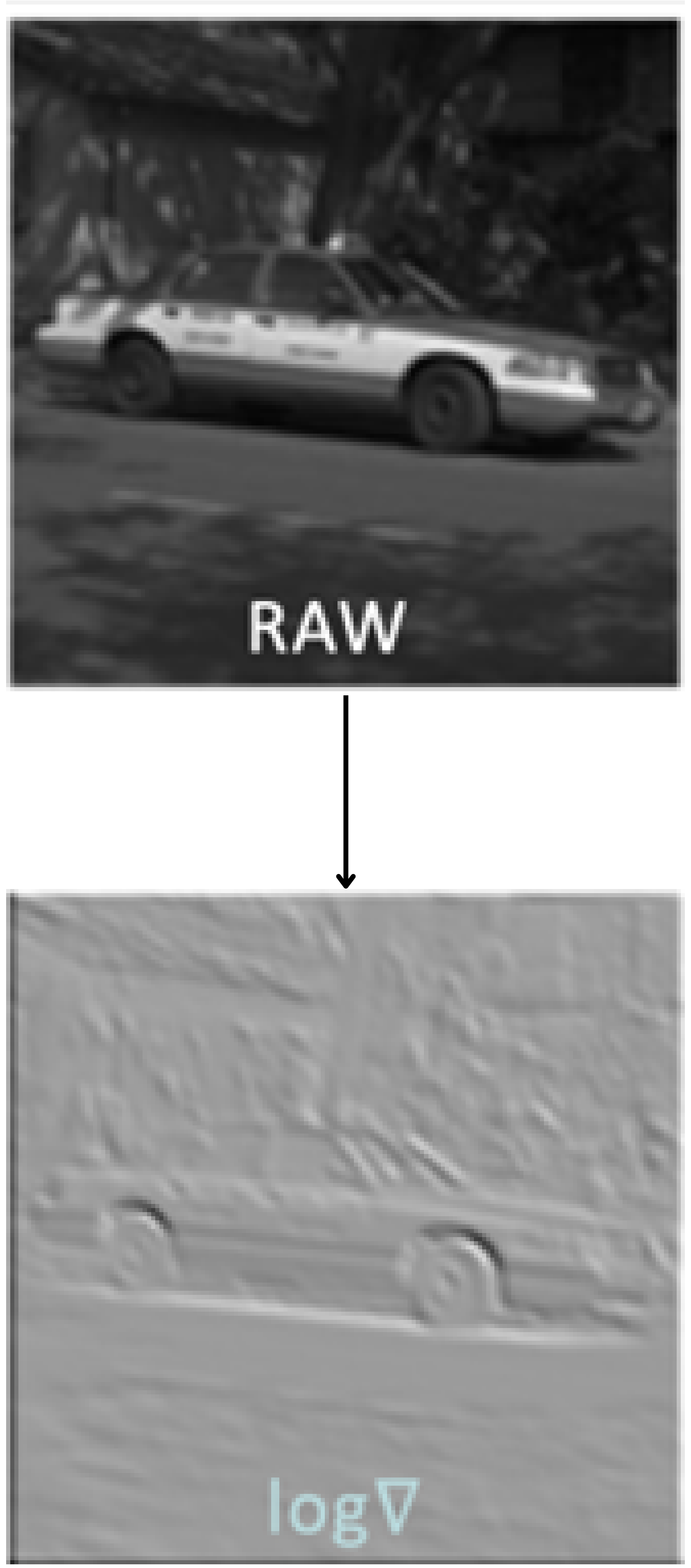
- Image $P \in \mathbb{R}^{H \times W}$

- $P' = \log(P+1)$ (normalize illumination variances)

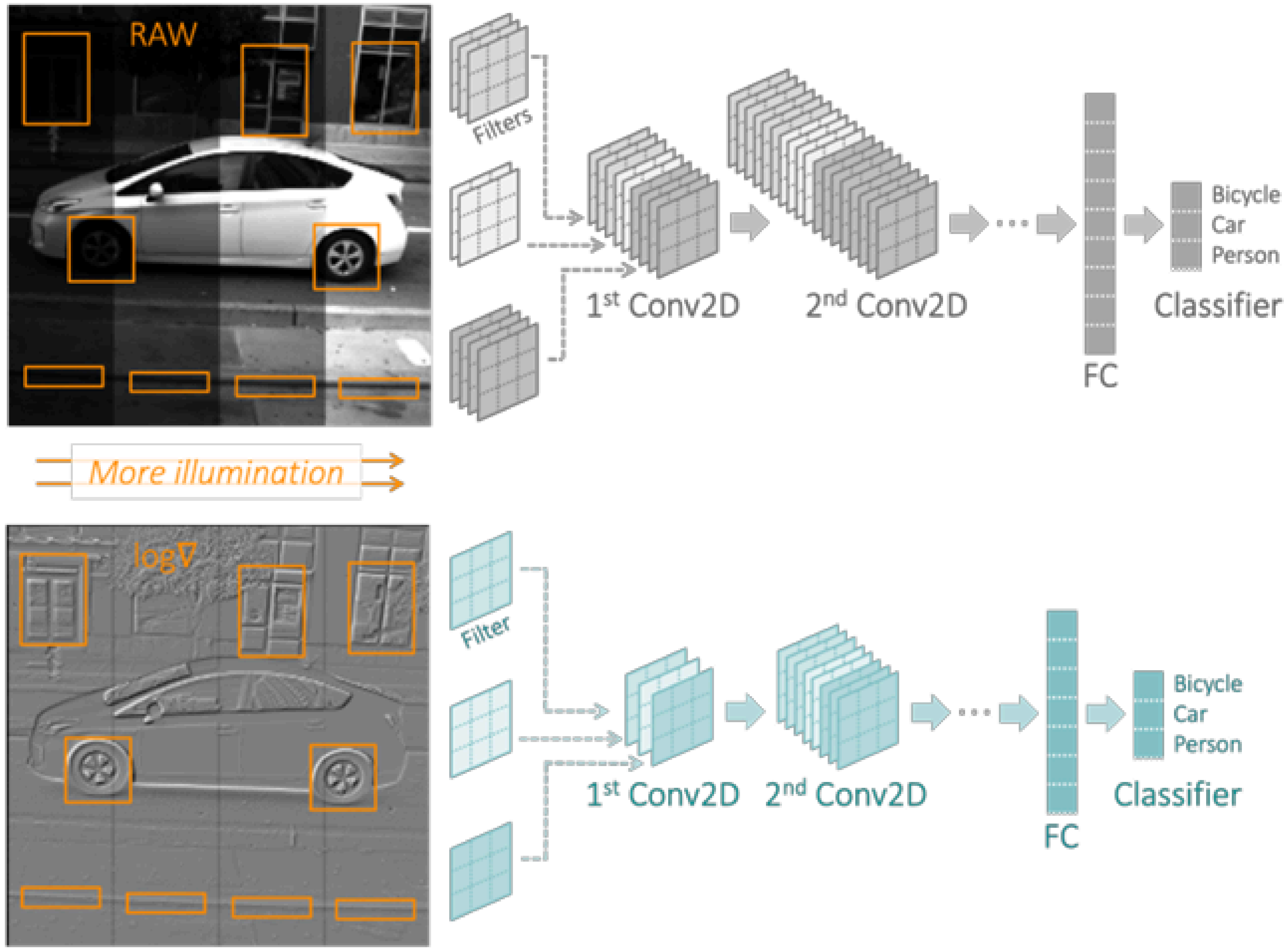
- $\log \nabla = P' * f$ where $f = \begin{bmatrix} 0 & -1 & 0 \\ -1 & 0 & 1 \\ 0 & 1 & 0 \end{bmatrix}$

- Ratio-to-digital converter (RDC)

$$(\log \nabla)_{j,k} = \log P_{j,k-1} - \log P_{j,k+1} = \log\left(\frac{P_{j,k-1}}{P_{j,k+1}}\right) \approx Q\left(\frac{P_{j,k-1}}{P_{j,k+1}}\right)$$



Log ∇ Intuition



$$\log\left(\frac{\cancel{a}\cdot P_{j,k-1}}{\cancel{a}\cdot P_{j,k+1}}\right)$$

illumination invariance

- **Robustness** to global illumination changes
- Better **quantization**
- Smaller CNNs

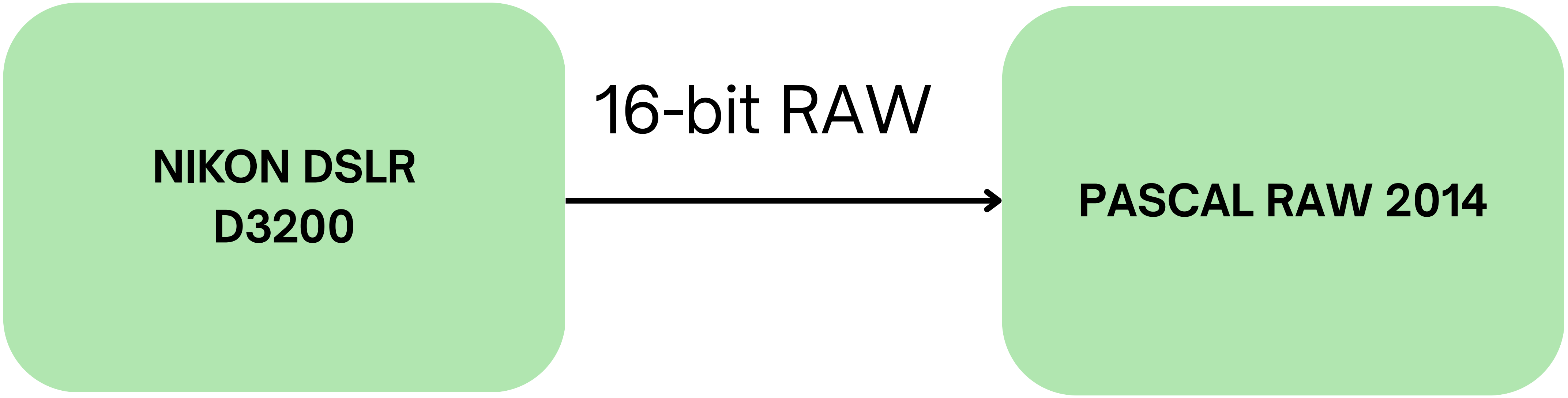
Log gradients remove the lighting and keep the structure and shape.

Datasets

PASCAL RAW 2014

- 6550 images - demosaicked grayscale
- 3 classes: bicycle, car, person
- It is a RAW dataset, meaning no ISP was applied
- Images are all from the same sensor
- Closer to the real-world scenarios

- Other datasets, like Visual Wake Words, were not used due to their processing of images. Usually given as JPEGs

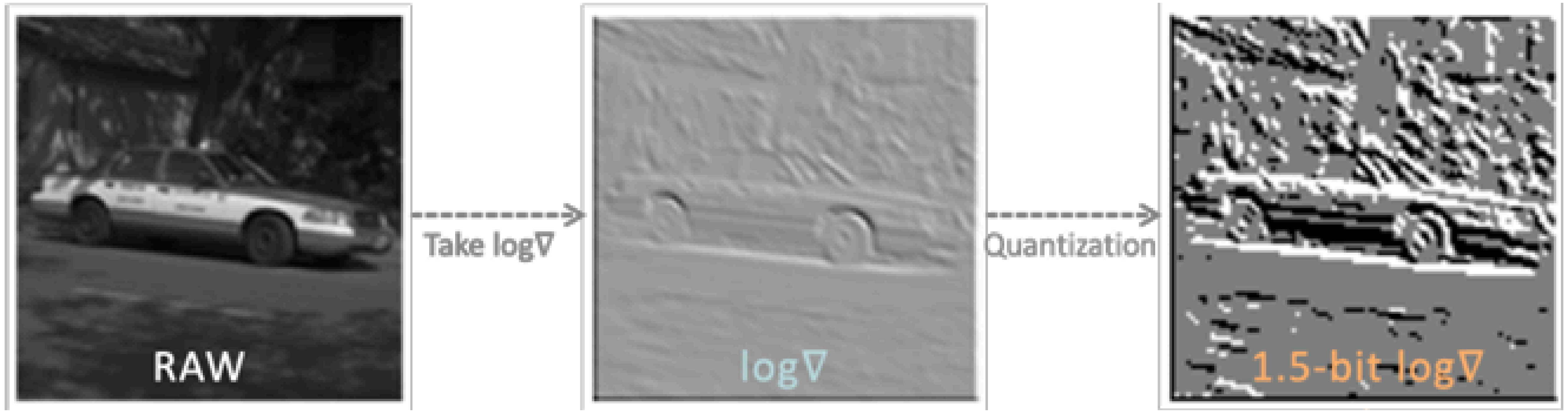


- Bicycle (708)
- Car (1765)
- Person (4077)

Experiements

Comparison of metrics for:

- 8-bit JPEG - generated from RAW images using only gamma correction
- 16-bit RAW - demosaiced grayscale images from PASCAL RAW 2014
- FP log∇ - no quantization
- 1.5-bit log∇ - 3 level quantization using empirical thresholds
- 2.25-bit log∇ - 5 level quantization using empirical thresholds

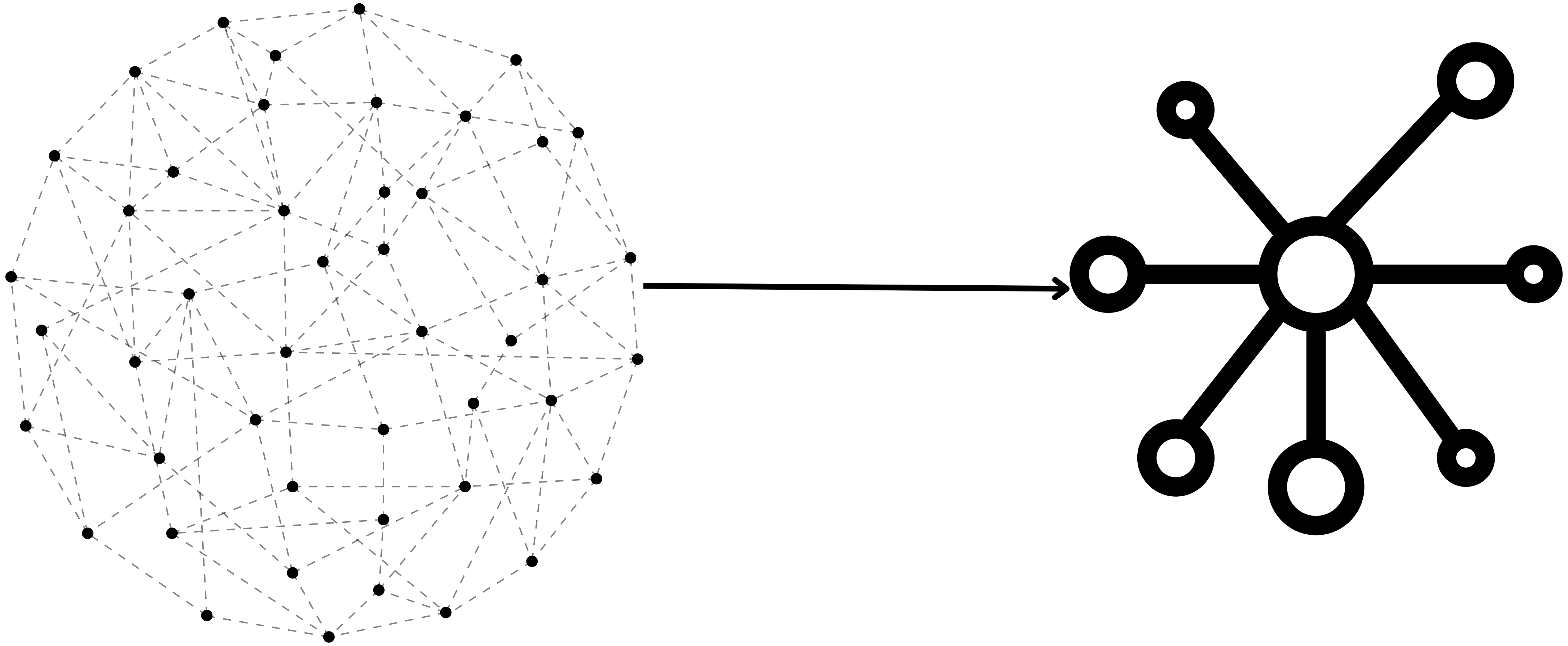


MAIN IDEA

Log∇ needs smaller CNNs

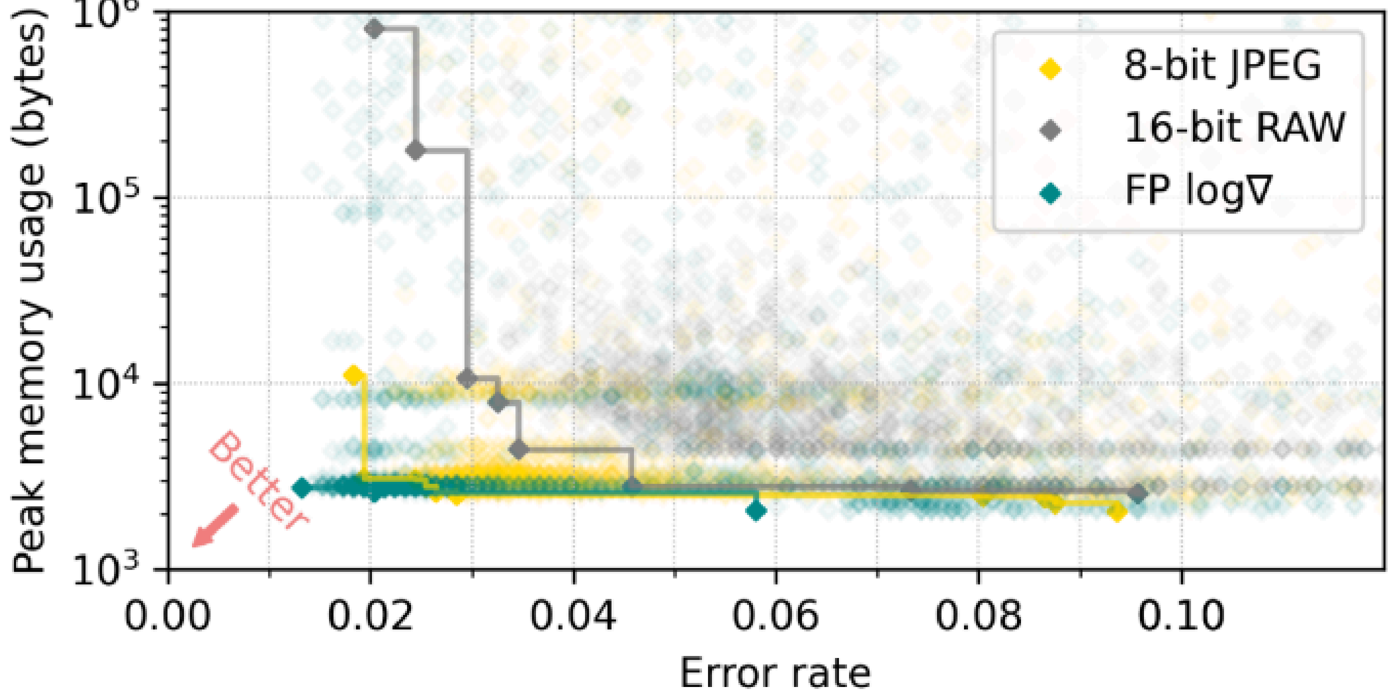
Architecture Search

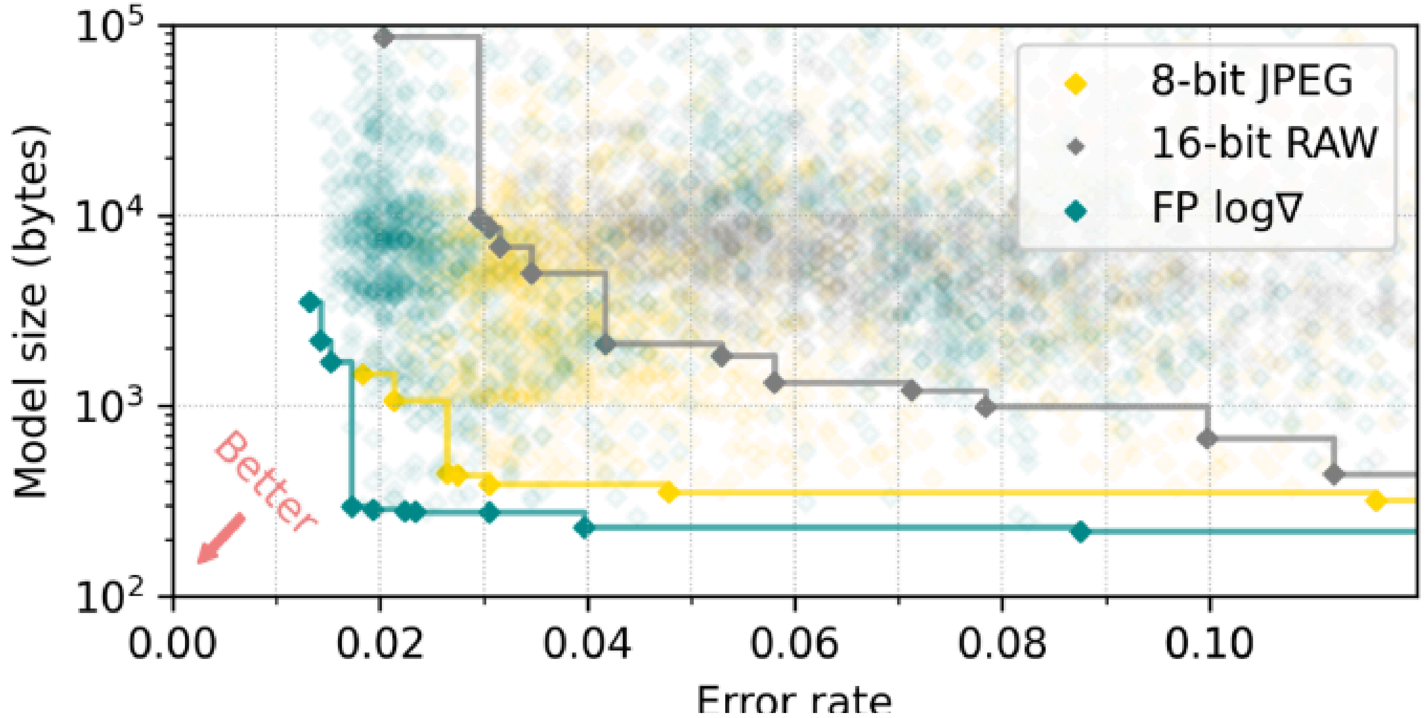
- μNAS (Microcontroller Neural Architecture Search): NAS algorithm made for resource-constrained environments (RAM, persistent storage, processor speed)
- Aging evolution: evolutionary algorithm that maintains diversity in the search population by replacing the oldest models
- Structural pruning: technique to remove redundant parts of the network, further reducing model size and complexity

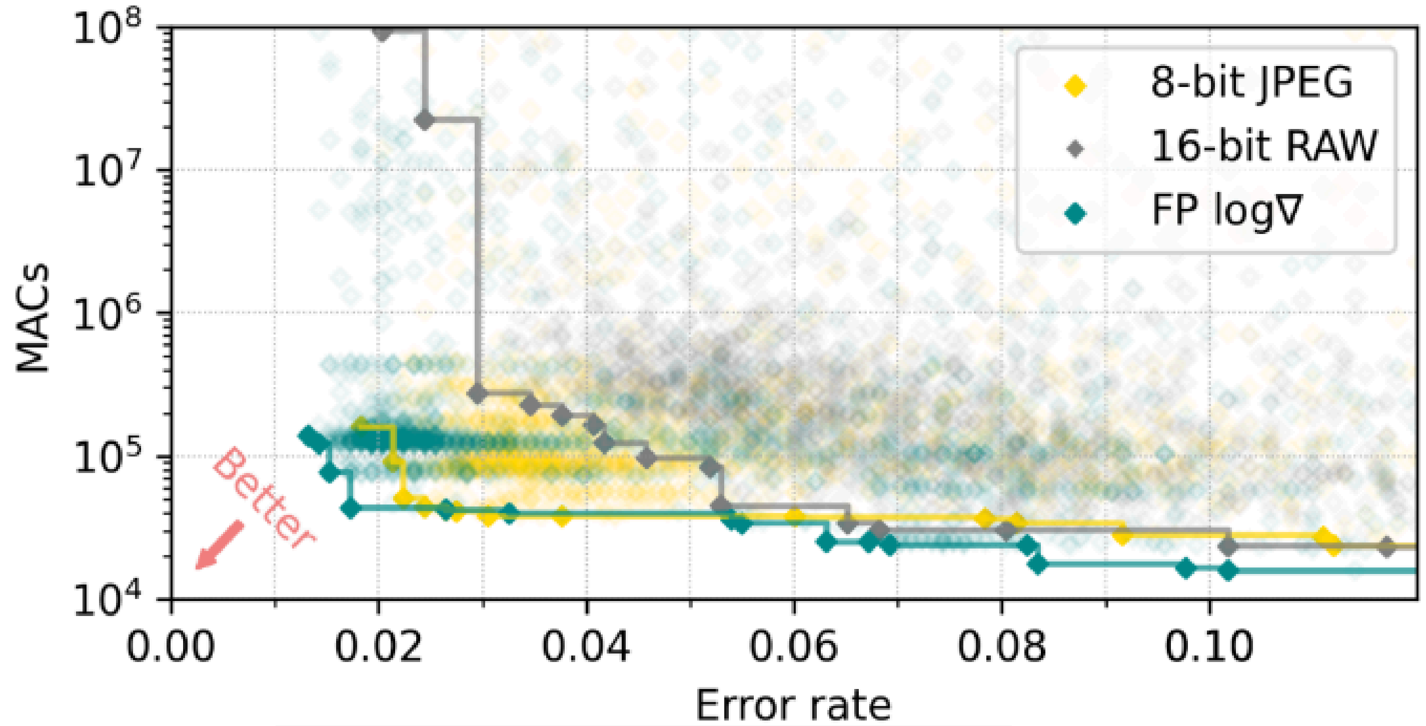


μNAS

- CNN architectures that are both accurate and efficient on microcontrollers
- **High granularity search space:** allows fine-grained control over architectural components, such as filter sizes, number of channels, and layer types
- **Minimal connectivity restrictions:** allows flexible layer connections, enabling the discovery of efficient architectures

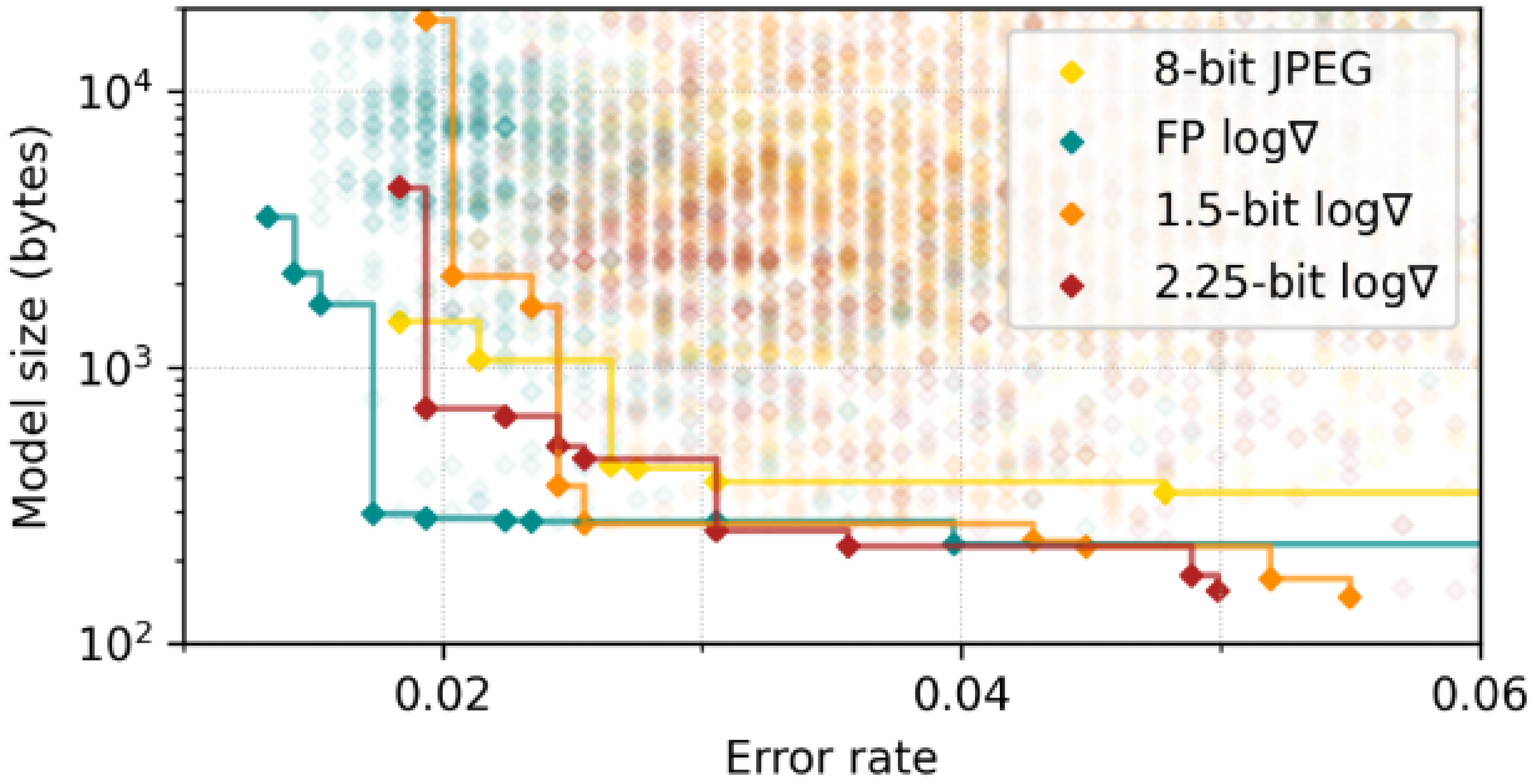


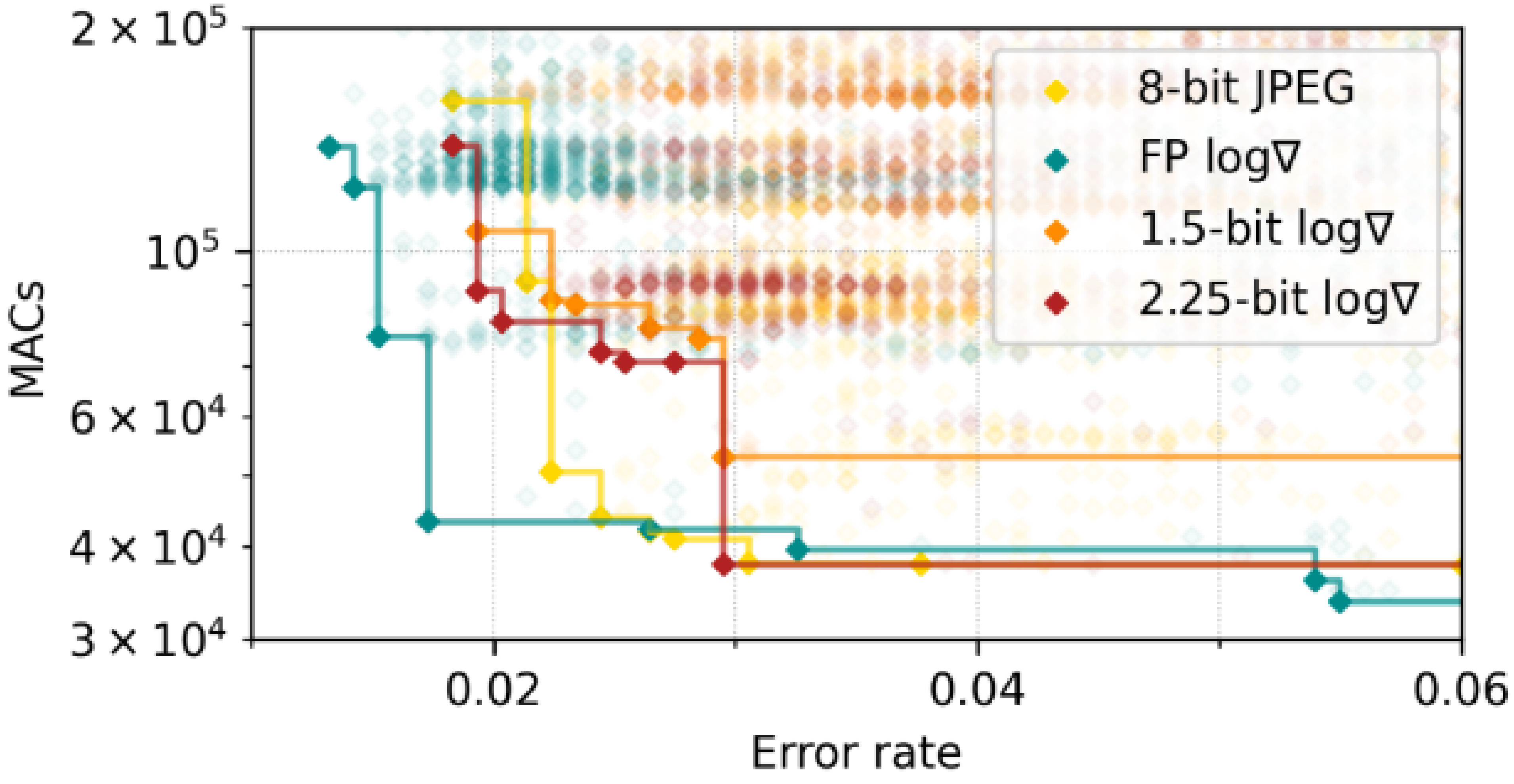


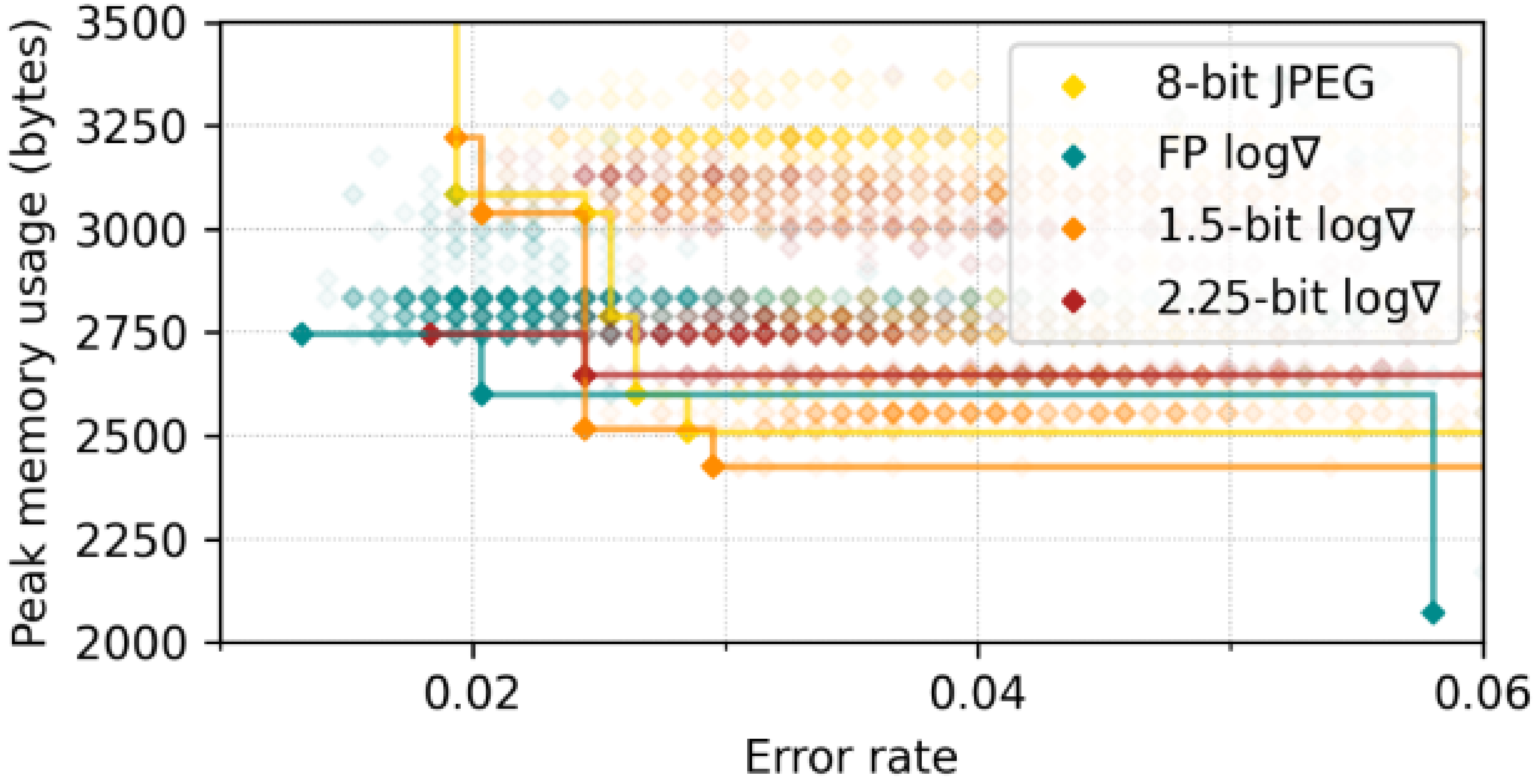


µNAS

Performance of quantized models



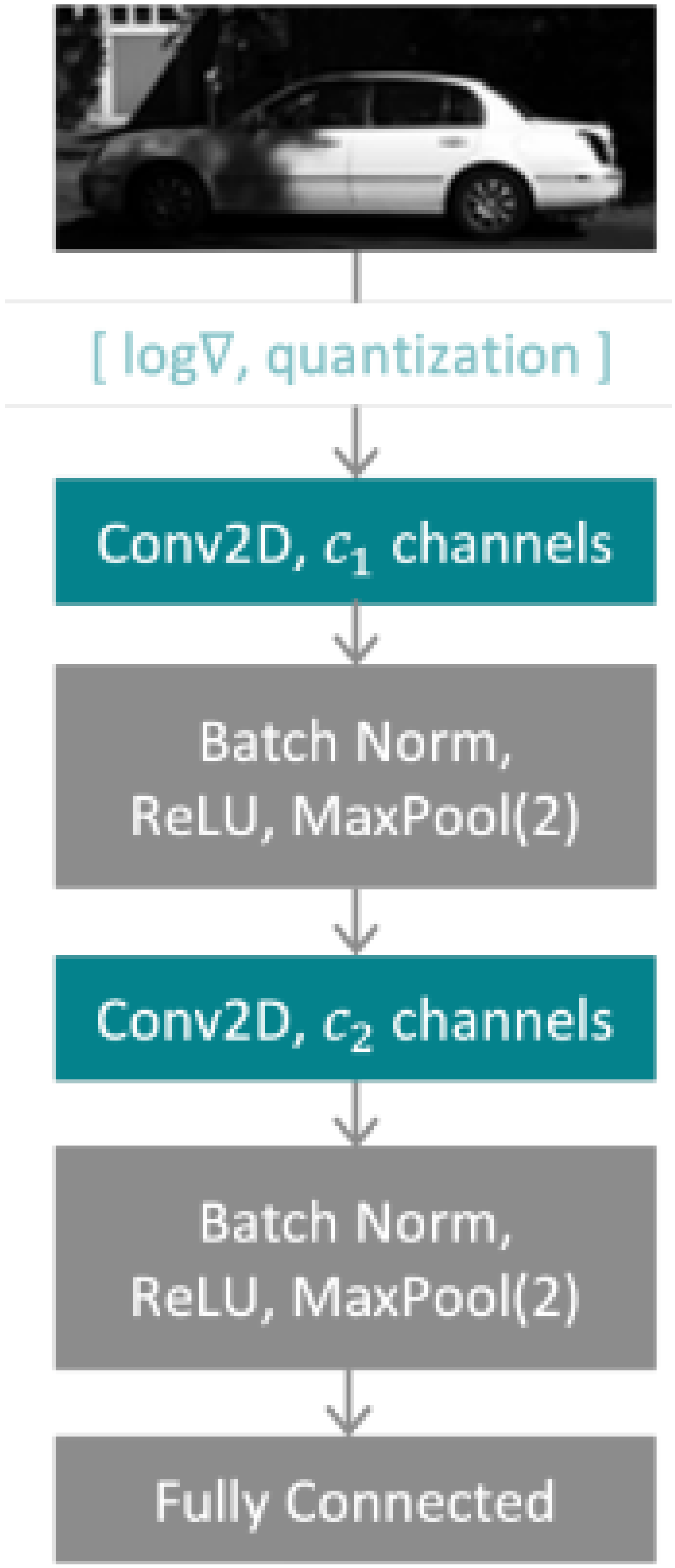




Each individual dot = one architecture candidate produced during the search process

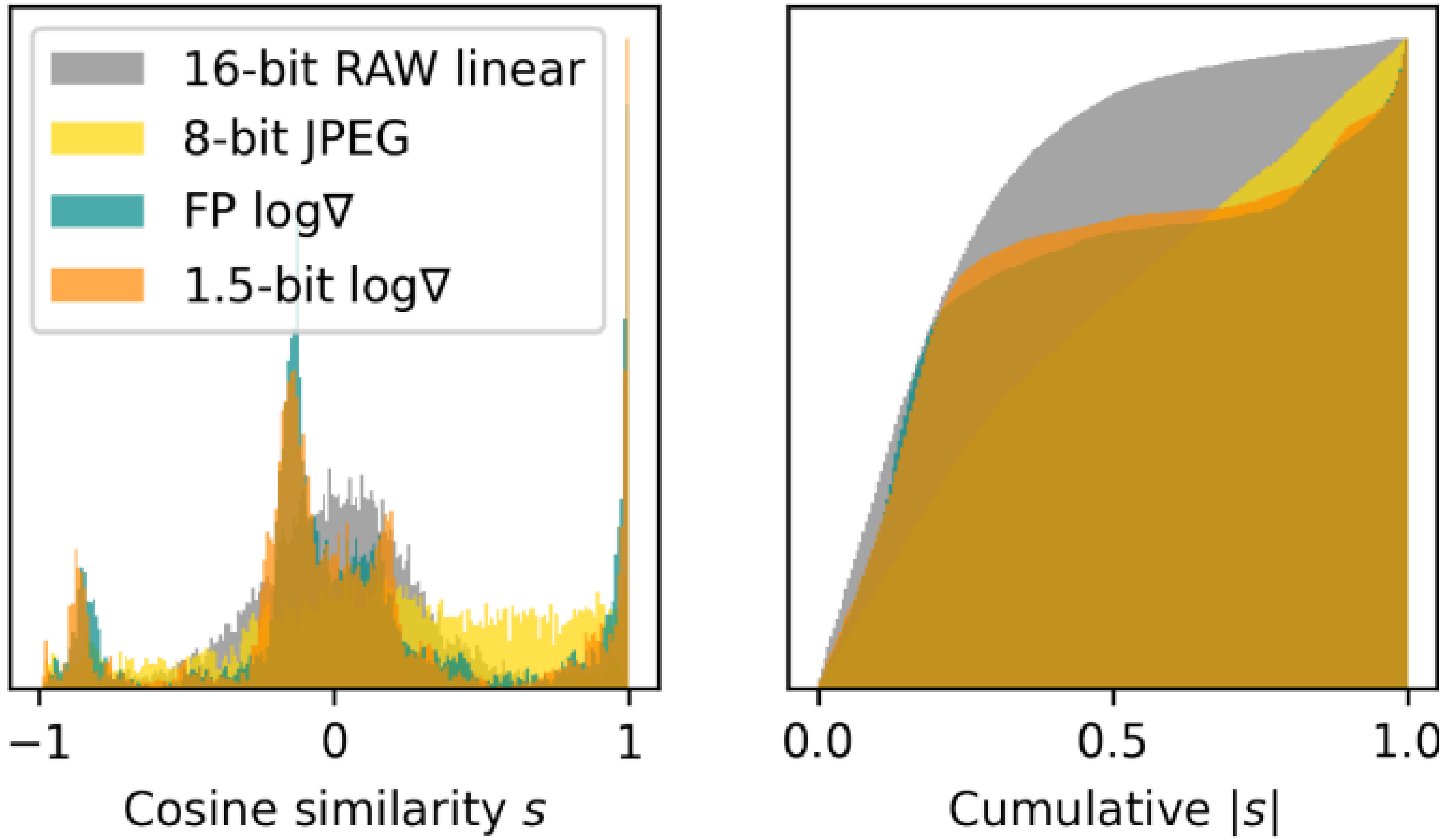
Fixed Architectures

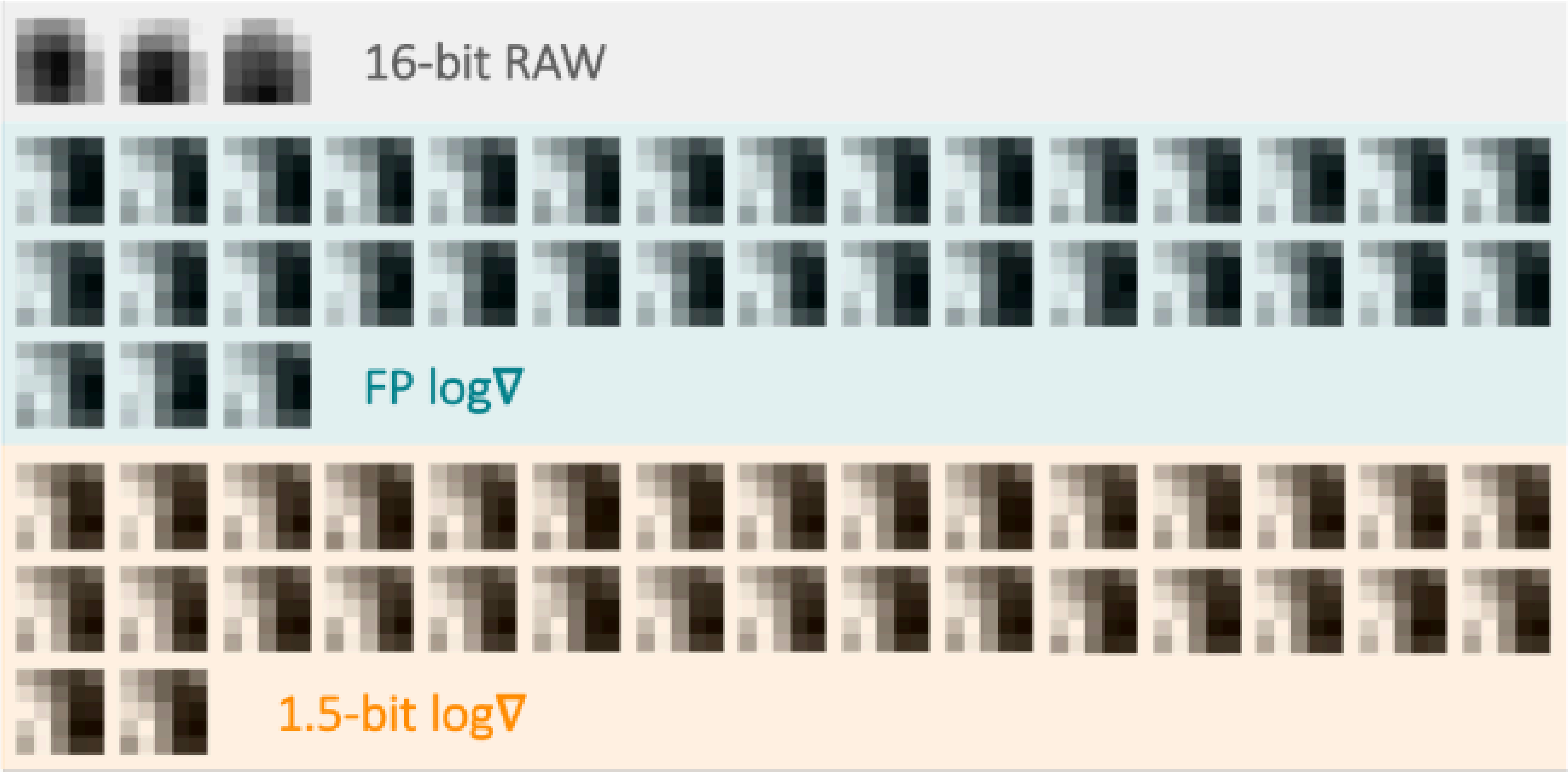
- Observed change in filter redundancy if we keep the CNN architecture fixed and vary the input
- c1 = 150, c2 = 5
- Higher filter redundancy means that there is a higher degree of similarity among the filters
- Leads to higher prunning possibilites



Fixed Architectures

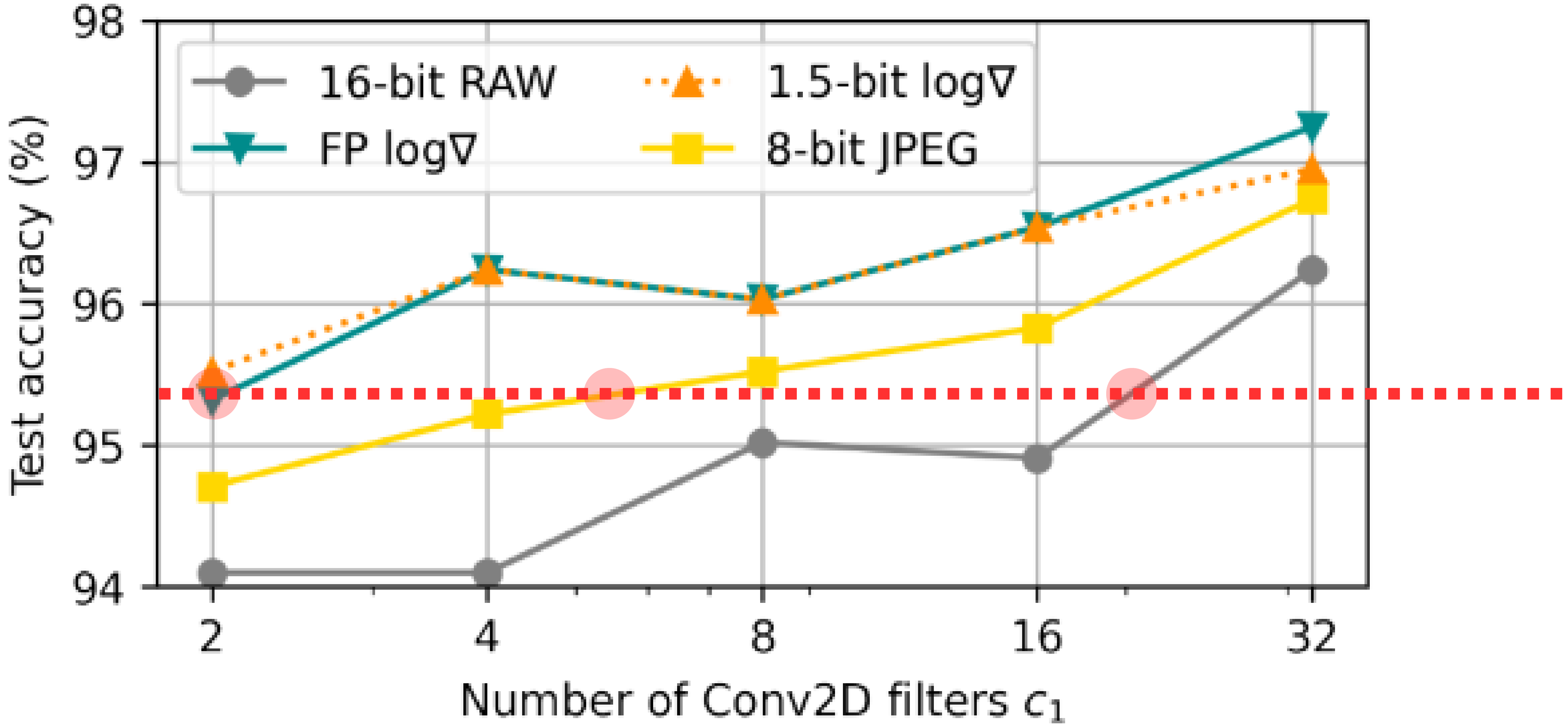
- Computer layer-wise **cosine similarities** among CNN filters after training
- Visualize similar filters for comparison





Fixed Architectures

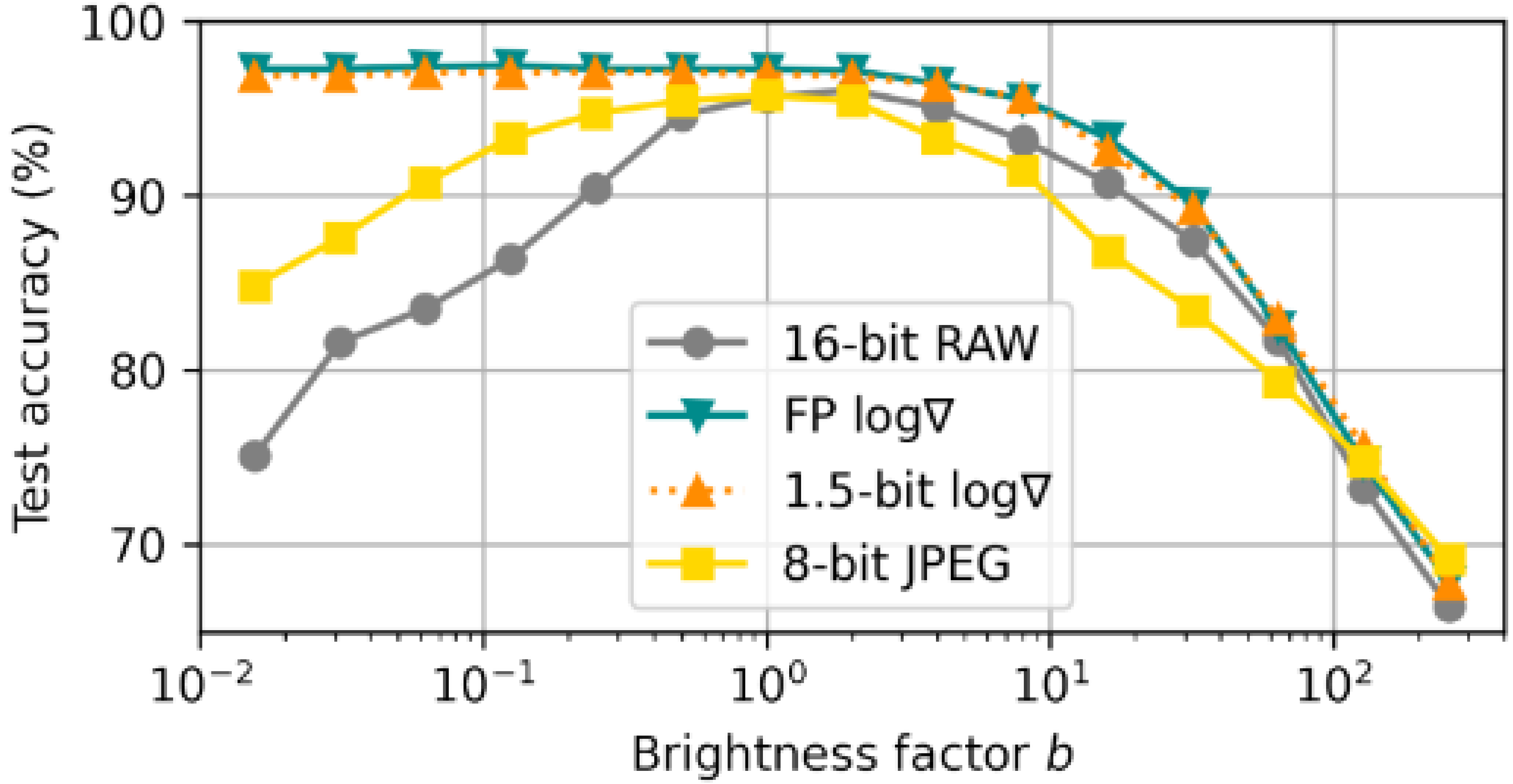
- RoyChowdhury et al. have shown that higher filter similarity allows more channel pruning
- Confirmed by fixing c2 = 8, and c1 ∈ {2, 4, 8, 16, 32}



log-∇ needs fewer filters in the 1st Convolutional Layer

Fixed Architectures

- Sensitivity to simulated illumination changes
- Largest networks from the previous experiment (c1 = 32, c2 = 8)
- Vary the brightness of test images by factor $b \in \{2^{-6}, 2^{-5}, \ldots, 2^{8}\}$ relative to the nominal training brightness

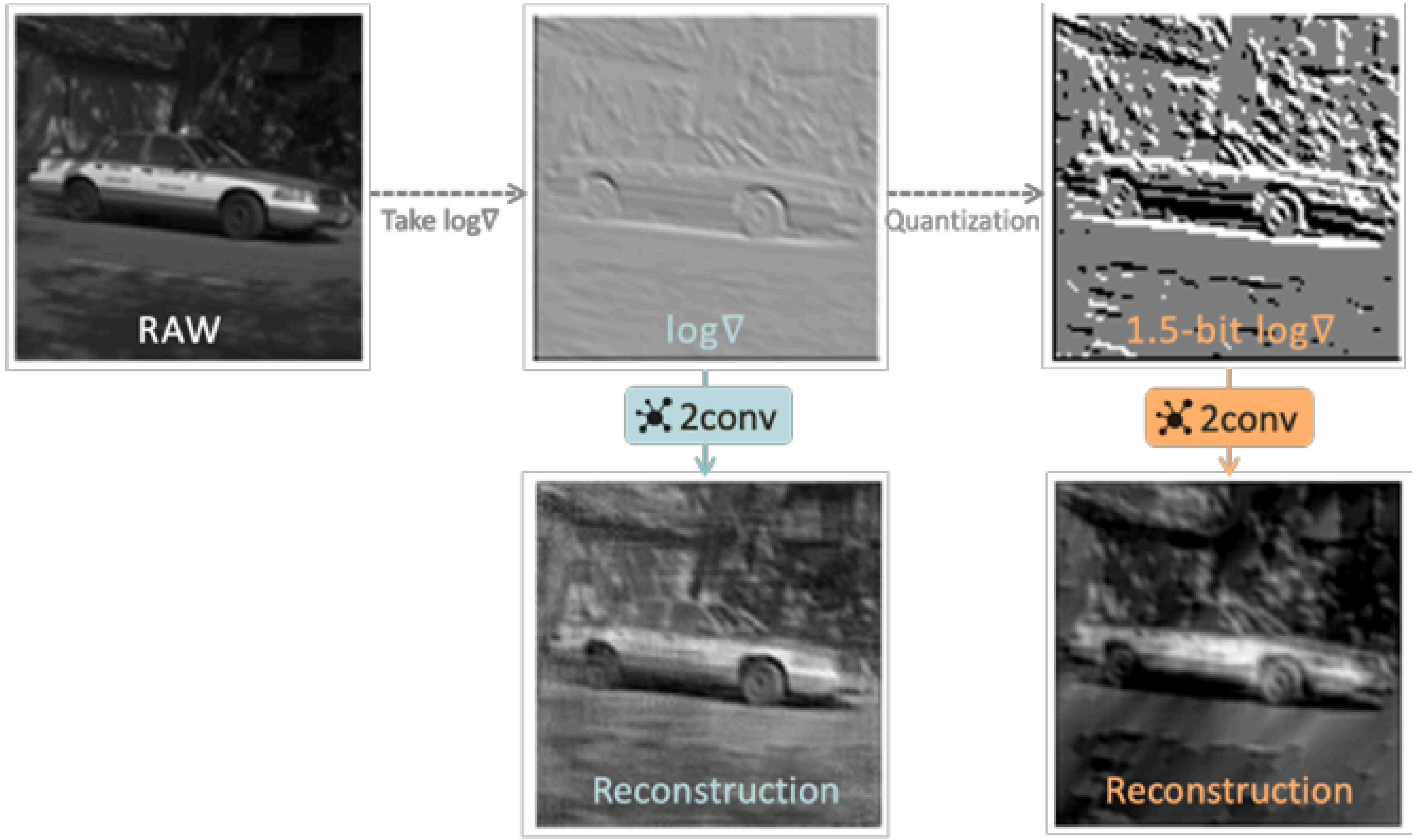


Conclusions

- Log gradient inputs make the CNN more compressible and less sensitive to illumination
- Quantization down to 1.5 bits for input
- Unprocessed RAW images used to obtain log gradients. JPEGs and other processed formats don’t represent real-world well
- Future work should consider training-based optimization of the log ∇ quantization thresholds, quantized training to reduce the internal compute precision of the CNN, as well as the response to the adversarial inputs

Thank you

APPENDIX


RAW
Take log∇
log∇
Quantization
1.5-bit log∇
2conv
2conv
Reconstruction
Reconstruction