Research Square

Preprints are preliminary reports that have not undergone peer review.
They should not be considered conclusive, used to inform clinical practice,
or referenced by the media as validated information.

Mitigating Resource-Constraints in Computer Vision Systems Using Deep, Sparse CNN Models

Mohamed Ibrahim Abdelaziz

melsherbiny0924@gmail.com

Independent researcher

Research Article

Keywords: Computer Vision, YOLO, Efficient Computing

Posted Date: November 3rd, 2026

DOI: https://doi.org/10.21203/rs.3.rs-11576281/v1

License: This work is licensed under a Creative Commons Attribution 4.0 International License.
Read Full License

Additional Declarations: The authors declare no competing interests.

Mitigating Resource-Constraints in Computer Vision Systems Using Deep, Sparse CNN Models

Mohamed Ibrahim Abdelaziz
Independent Researcher
https://github.com/Mohamed2Elsherbiny/epfl-sparse-cnn-detection

Abstract

Resource-constrained deployment of object detectors demands models that are simultaneously accurate, robust, and computationally efficient. We investigate whether biologically-inspired activation sparsity—where only a fraction of neurons fire per inference, as in mammalian neocortex—can address all three requirements through a single mechanism. We inject a k-Winner-Take-All (kWTA) activation (retaining only the top-r fraction of channels per spatial location with homeostatic boosting) into the YOLOv11n backbone. Comparing an r=0.5 sparse backbone against an identically configured dense baseline (both trained from scratch, 30 epochs, PASCAL VOC), we find that: (1) with kWTA active at inference, the sparse model substantially outperforms the dense baseline (mAP@0.50: 0.704 vs. 0.600, +17.3% relative) while simultaneously zeroing 50% of backbone activations, (2) sparse-trained representations are more robust to Gaussian noise, blur, and occlusion (+29.6% relative mean corrupted mAP@0.50, 15/15 conditions favoring the sparse model), (3) an estimated 48.5% of backbone MACs become eliminable under sparsity-aware execution, and (4) both models exhibit negligible degradation under INT8 post-training quantization (<0.001 mAP loss). These results suggest that competitive activation sparsity can simultaneously deliver accuracy improvement *and* activation-level computational reduction, offering a promising direction for resource-constrained edge deployment.

1 Introduction

Deploying object detection models on resource-constrained platforms—mobile devices, drones, medical imaging appliances, autonomous vehicles—faces three compounding challenges: (i) limited computational budgets requiring fewer floating-point operations, (ii) vulnerability to sensor noise and environmental degradation in real-world conditions, and (iii) the need for aggressive quantization (INT8/FP16) to fit within memory and latency envelopes. Current approaches address these independently: efficient architectures reduce FLOPs [12, 19], robustness training augments data with corruptions [11], and quantization-aware training minimizes bit-width degradation [13]. No single mechanism addresses all three simultaneously.

The mammalian neocortex suggests an alternative. Despite operating with $\sim 10^{11}$ neurons on millisecond timescales, the brain achieves remarkable efficiency through *sparse distributed representations* (SDRs)—at any moment, only 1–4% of neurons are active [10]. Hierarchical Temporal Memory (HTM) theory formalizes this insight: competitive inhibition and homeostatic plasticity produce sparse codes that are simultaneously efficient (fewer active units), robust (distributed encoding tolerates noise), and information-dense (high combinatorial capacity) [2, 4].

We ask: *can a single HTM-inspired sparsity mechanism, injected into a modern object detector's backbone, simultaneously improve accuracy, robustness, and activation-level computational efficiency?*

We inject k-Winner-Take-All (kWTA) activations—retaining only the top-r fraction of channels with homeostatic boosting—into the YOLOv11n backbone during training. Our controlled comparison (dense vs. r=0.5 sparse, both from scratch, seed 42, 30 epochs, PASCAL VOC 2007+2012) reveals that this single mechanism:

1. Creates a **co-adapted architecture**: when kWTA remains active at inference, the sparse-trained model outperforms the dense baseline by +17.3% relative mAP@0.50 (0.704 vs. 0.600), meaning accuracy and activation-level efficiency are achieved simultaneously.

2. Confers **corruption robustness**: +29.6% relative mean corrupted mAP@0.50 across noise, blur, and occlusion (15/15 conditions favoring the sparse model).

3. Enables **estimated computational savings**: an estimated 48.5% of backbone MACs become eliminable under sparsity-aware execution engines, with the accuracy gain preserved.

4. Maintains **quantization compatibility**: both models show negligible INT8 degradation (<0.001 mAP loss), confirming orthogonal deployability with post-training quantization.

The remainder of this paper presents related work (Section 2), our methodology (Section 3), experimental results including five resource-oriented analyses (Section 4), discussion of implications for constrained deployment (Section 5), and conclusions (Section 6).

2 Related Work

Efficient object detection. The YOLO family [18, 14] achieves real-time detection through single-stage design. Complementary efficiency approaches include depthwise separable convolutions [12, 19], channel shuffling [21], and neural architecture search [20]. These reduce per-operation cost; our approach reduces the *number of operations* by zeroing activations.

Sparsity in neural networks. Weight pruning [8, 5] removes connections post-training. Activation sparsity [6, 17, 15] dynamically zeros units based on input. The k-sparse autoencoder [17] and soft threshold reparameterization [15] operate in classification; their extension to detection backbones is unexplored.

Activation sparsity for vision efficiency and robustness. Recent work has demonstrated that activation sparsity can accelerate vision models in practice. Grimaldi et al. [7] show that semi-structured activation sparsity yields measured inference speedups for object detection on GPU hardware. SparseViT [3] applies window-level activation pruning to vision transformers with latency reductions on detection tasks. On the robustness side, Luo et al. [16] demonstrate that Top-K activation sparsity in CNNs induces shape-bias emergence, linking sparse activations to improved corruption robustness. Our work differs in mechanism (HTM-inspired homeostatic competitive selection rather than semi-structured patterns or learned thresholds) and in scope (joint characterization of accuracy, robustness, and sparsity within a single YOLO backbone modification).

HTM and biological sparsity. Hierarchical Temporal Memory [10, 9, 4] formalizes cortical sparse coding. Ahmad and Hawkins [1] demonstrated that kWTA activations achieve competitive classification accuracy with dramatically fewer active units. However, their work addresses classification only; the spatial precision requirements of bounding-box regression in detection remain uncharacterized.

Research gap. While activation sparsity has been explored for efficient inference [7, 3] and robustness [16] independently, no prior work investigates whether *HTM-inspired homeostatic* kWTA—with its biologically-motivated duty-cycle regulation and competitive boosting—can jointly provide

accuracy improvement, corruption robustness, and activation-level efficiency within a single object detection backbone. We address this specific gap.

3 Methodology

3.1 k-Winner Activation with Homeostatic Boosting

Given input $x \in \mathbb{R}^{B\times C\times H\times W}$, the k-Winner activation retains only the top-k channels per spatial location:

$$y_{c,h,w} = \begin{cases} x_{c,h,w} & \text{if } x_{c,h,w} \geq \tau_{h,w} \\ 0 & \text{otherwise} \end{cases}, \quad k = \lfloor r \cdot C \rfloor \tag{1}$$

where $\tau_{h,w}$ is the k-th largest value across channels at position (h, w), and $r \in (0, 1)$ is the retention ratio (we use r=0.5, giving 50% sparsity).

Homeostatic boosting prevents dead channels through duty-cycle tracking:

$$d_c^{(t+1)} = 0.99 \cdot d_c^{(t)} + 0.01 \cdot \bar{a}_c, \quad b_c = \exp\left(-\beta\left(\frac{d_c}{r} - 1\right)\right) \tag{2}$$

where $\bar{a}_c$ is the mean activation indicator for channel c and b_c scales the pre-competition input. Under-active channels ($d_c < r$) receive $b_c > 1$, increasing their chance of winning.

Gradient flow uses the straight-through estimator: $\partial\mathcal{L}/\partial x = (\partial\mathcal{L}/\partial y) \cdot \texttt{sg}(m)$, where m is the binary winner mask and `sg` denotes stop-gradient.

3.2 Architecture: Sparse YOLOv11n Backbone

We inject kWTA activations into YOLOv11n's backbone only (layers 0–10, replacing 35 SiLU activations), preserving the neck and Detect head unchanged. This isolates sparsity as the sole experimental variable. The model has 2,609,500 parameters regardless of sparsity (kWTA alters activations, not weights).

Sparsity at inference. Our sparse inference experiments (Section 4.4) reveal that the sparse-trained model requires kWTA to remain active at inference to achieve its full accuracy. When Ultralytics' layer fusion strips kWTA (replacing it with SiLU), accuracy drops below the baseline. This demonstrates that the weights are co-adapted with the competitive activation pattern—a structural dependency that has practical implications for deployment. On platforms supporting sparse activation operators, this dependency is beneficial: the accuracy gain and 50% channel sparsity are inherently coupled, so both are realized simultaneously. However, on platforms that cannot preserve the kWTA gate, the model cannot be deployed as a standard dense network without accuracy degradation.

3.3 Training Protocol

Both models (dense baseline, sparse r=0.5) are trained identically from scratch: SGD (lr=0.01, momentum=0.937, weight decay 5×10^{-4}), cosine LR with 3-epoch warmup, 30 epochs, batch 16, 640×640 input, mosaic augmentation, seed 42, on NVIDIA T4 GPU. Training data: VOC 2007+2012 trainval (16,551 images). Evaluation: VOC 2007 test (4,952 images).

4 Experiments and Results

4.1 Main Detection Performance

Table 1 presents the core finding: the sparse-trained model achieves its best performance when kWTA remains active at inference (Mode C), substantially outperforming the dense baseline.

Table 1: Detection performance on VOC 2007 test. The sparse model achieves its best accuracy *with kWTA active at inference* (Mode C), directly coupling the accuracy gain with activation-level sparsity.

Model	**Params**	**mAP@.50**	**mAP@.50:.95**	**Train (h)**
Baseline (dense, scratch)	2.61M	0.600	0.398	1.19
HTM Sparse r=0.5 — sparse inf.	2.61M	**0.704**	**0.500**	2.56
HTM Sparse r=0.5 — dense inf.	2.61M	0.568	0.377	2.56
Rel. improv. (sparse vs. baseline)	—	**+17.3%**	**+25.6%**	2.15×

A key observation: the sparse-trained model requires kWTA to remain active at inference to realize its full accuracy. When layer fusion strips the kWTA gates (dense inference, Mode B), accuracy drops to 0.568 mAP@0.50—below the baseline. This demonstrates that the learned weights are co-adapted with the competitive activation pattern: kWTA is not merely a training regularizer but an *integral architectural component*. This co-adaptation has a dual character. On one hand, it constitutes a deployment constraint—the model cannot be run as a standard dense network without accuracy loss. On the other hand, it ensures that the computational savings (50% fewer active backbone channels) and the accuracy improvement are *inherently coupled*: any deployment that preserves kWTA simultaneously realizes both benefits.

4.2 Robustness to Input Corruptions

Real-world edge deployment exposes detectors to degraded inputs. We evaluate both checkpoints on VOC 2007 test under three corruptions at five severity levels (15 conditions total):

Table 2: Robustness evaluation (mean mAP@0.50 across severity levels 1–5).

Model	**Clean**	**Noise**	**Blur**	**Occl.**	**Mean Corr.**	**Rel. Drop**
Baseline	0.600	0.298	0.226	0.190	0.238	−60.4%
HTM r=0.5	0.704	0.376	0.317	0.231	0.308	−56.2%
Rel. improvement	+17.3%	+26.4%	+40.5%	+21.6%	**+29.6%**	+4.2pp

The sparse model achieves higher mAP in all 15 conditions. Descriptive paired statistics across the 15 corruption cells: mean mAP@0.50 difference +0.070 (95% bootstrap CI [0.048, 0.093]), Wilcoxon signed-rank $p \approx 6\times10^{-5}$. We emphasize that these statistics describe a single checkpoint pair (seed 42); they quantify the consistency of the advantage across corruption conditions but do not constitute population-level inference over model initializations.

4.3 Computational Efficiency Analysis

To characterize the potential computational benefit of activation sparsity, we measure the sparse model's activation profile and estimate the resulting MAC reduction under an idealized sparsity-aware

execution model:

Table 3: Computational efficiency: backbone analysis (measured on VOC 2007 test).

Metric	**Baseline**	**HTM** r=0.5
Mean backbone activation sparsity	0.01%	24.6%
Theoretical backbone MACs (dense)	1.52 GMACs	1.52 GMACs
Estimated backbone MACs (sparse-aware)	1.52 GMACs	0.78 GMACs
Estimated MAC reduction	—	48.5%
Estimated memory bandwidth saving (FP32/image)	—	18.8 MB
Inference latency (CPU, batch=1)	125.1 ms	125.8 ms
Parameters	2,609,500	2,609,500

With 24.6% measured backbone activation sparsity (lower than the theoretical 50% due to measurement at Conv outputs before the kWTA gate), sparsity-aware execution engines could in principle skip approximately 48.5% of backbone MACs. The identical dense-inference latency (125 ms on CPU) confirms that *without specialized sparse hardware or kernels*, the computational savings remain theoretical. Translating activation zeros into real latency or energy reductions requires runtime support for sparse computation [7]—a hardware capability that is increasingly available but not yet validated for our specific activation pattern. The accuracy benefit, however, is immediately realizable on any platform that preserves the kWTA operator.

4.4 Sparse Inference Evaluation

A key question for deployment: does keeping sparsity active at test time compound or degrade the accuracy benefit? We compare four inference modes on VOC 2007 test:

Table 4: Sparse vs. dense inference on both weight sets (VOC 2007 test).

Inference Mode	**mAP@0.50**	**mAP@0.50:0.95**
(A) Dense inference → dense weights	0.600	0.398
(B) Dense inference → sparse weights	0.568	0.377
(C) Sparse inference → sparse weights	**0.704**	**0.500**
(D) Sparse inference → dense weights	0.070	0.038

Mode C demonstrates that the sparse-trained model achieves its full accuracy *with sparsity active at inference*, meaning the 50% activation reduction is realizable at deployment time without sacrificing accuracy. Mode B confirms that removing kWTA at inference degrades accuracy below the dense baseline—the model cannot simply be deployed as a conventional dense network. Mode D confirms that inference-time sparsity alone (without sparse training) destroys performance—the benefit requires training-time adaptation. Together, Modes B–D establish that kWTA creates a bidirectional co-adaptation: the weights depend on kWTA, and kWTA depends on sparse-trained weights. A r-ratio sweep at inference further confirms strong operating-point specificity: r=0.5 is optimal (matching the training ratio); r=0.3 degrades to 0.372 mAP@0.50, while $r \geq 0.7$ collapses entirely, indicating that the network has learned activation distributions specifically calibrated to 50% retention.

4.5 Per-Class Detection Analysis

We extract per-class AP@0.50 for all 20 VOC categories under sparse inference (Mode C). The sparse model achieves its best results with kWTA active:

- Under sparse inference, the model improves AP in all 20 categories compared to the dense baseline, with gains ranging from +0.04 (pottedplant) to +0.12 (cat, dog).
- Largest improvements are observed for categories requiring fine-grained feature discrimination (animals, vehicles), consistent with the competitive feature-selection hypothesis.
- The improvement is general across object sizes (small, medium, large), confirming the regularization benefit is not category-specific.

4.6 Feature Representation Analysis

To understand *why* sparse training produces better generalization, we analyze internal representations at backbone stages using 64 VOC 2007 test images:

Effective rank (SVD-based information dimensionality): We compute

$$\text{EffRank} = \exp\left(-\sum_i \hat{\sigma}_i \log \hat{\sigma}_i\right)$$

where $\hat{\sigma}_i$ are normalized singular values of the feature matrix. The sparse model achieves slightly higher effective rank at deeper stages (175.4 vs. 171.5 at stage3; 170.9 vs. 168.0 at stage4), indicating marginally more distributed information encoding.

Inter-channel correlation: The sparse model shows lower mean absolute inter-channel correlation at the final backbone stage (0.202 vs. 0.241), confirming that competitive activation produces less redundant feature channels where it matters most—at the representation fed to the detection neck.

CKA similarity: Centered Kernel Alignment between the two models decreases through the network (0.93 at stem $\rightarrow$ 0.68 at stage4_out), indicating that sparse training progressively reshapes representations, with the largest divergence at the deepest backbone stage. This confirms that kWTA's competitive dynamics accumulate a substantially different encoding strategy over depth.

4.7 Quantization Compatibility

Edge deployment typically requires INT8 quantization. We analyze whether sparse training affects quantization robustness by comparing weight distributions:

Table 5: Weight distribution analysis (backbone Conv layers): quantization-relevant metrics.

Metric	**Baseline**	**HTM** r=0.5	**Note**
Weight range (max−min)	2.078	2.598	Baseline narrower
Weight std	0.0433	0.0468	Baseline narrower
Outlier ratio ($> 3\sigma$)	1.27%	1.51%	Baseline fewer
Excess kurtosis	13.0	28.3	Baseline lower
INT8 mAP@0.50 degradation	−0.0001	−0.0002	Both negligible

Interestingly, the sparse model exhibits a wider weight range and higher kurtosis—arising from competitive training dynamics that create a few high-magnitude "specialist" weights—yet both models show negligible INT8 quantization degradation (<0.001 mAP loss). This indicates that YOLOv11n-scale models are inherently robust to post-training INT8 quantization regardless of training regime. The practical implication is positive: sparse training does not harm quantization deployability, meaning the kWTA accuracy and sparsity benefits compose orthogonally with standard INT8 deployment pipelines.

4.8 Stability Boundaries

We characterize the practical operating envelope: r=0.5 (50% retention) trains stably from scratch; $r \leq 0.2$ (80%+ sparsity) fails to converge; pretrained weight injection collapses entirely (mAP@0.50:0.95 $\approx$ 0.02). These boundaries establish where HTM-inspired sparsity is viable for detector training and motivate graduated curriculum approaches for future work.

5 Discussion

5.1 A Unified Mechanism for Multiple Constraints

Our central finding is that a single biologically-inspired mechanism—competitive activation sparsity—simultaneously addresses three distinct deployment challenges in this controlled setting:

1. **Accuracy with activation-level sparsity**: The sparse model achieves 17.3% higher relative mAP@0.50 than the dense baseline *while using only 50% of backbone channels at each inference.* Accuracy and activation-level efficiency are co-achieved rather than traded off.

2. **Robustness without augmentation overhead**: The +29.6% robustness improvement emerges from the sparse activation pattern itself. Features that survive competitive selection appear more noise-tolerant, reducing reliance on expensive corruption augmentation.

3. **Sparsity-ready activations**: Unlike training-only regularizers (dropout, weight decay), kWTA produces a model that *requires* its sparse activation at inference. This co-adaptation means that on platforms supporting sparse execution, the guaranteed 50% activation zeros are directly exploitable for MAC and energy savings. However, translating this activation sparsity into measured latency improvements requires sparse hardware or runtime support, which we have not validated.

5.2 Connection to Biological Efficiency

The brain achieves energy efficiency through sparse coding (few active neurons per input), competitive inhibition (winner-take-all dynamics), and homeostatic regulation (maintaining balanced utilization). Our kWTA mechanism replicates these principles, and the observed accuracy and robustness benefits are *consistent with* the hypothesis that these evolved properties confer computational advantages in artificial systems. However, we note an important caveat: our current experiments do not isolate the contribution of the HTM-specific components (homeostatic boosting, duty-cycle regulation) from the simpler Top-K competition mechanism. It is possible that a plain Top-K activation (without homeostatic boosting) would produce similar benefits. Disambiguating this requires an ablation comparing: (a) plain Top-K without boosting, (b) kWTA with homeostatic boosting (our method), and (c) random channel masking at 50%. Until such an ablation is conducted, the biological

motivation should be understood as an inspiration for the mechanism design rather than a confirmed causal explanation for the observed gains.

5.3 Comparison with Alternative Regularizers

Unlike dropout (random suppression), kWTA implements *competitive, input-dependent* selection. This is functionally richer: the same channel may be active for one input and suppressed for another, depending on relative activation magnitudes. Unlike weight decay (global magnitude penalty), kWTA operates on the activation manifold directly, shaping *which features are expressed* rather than *how large weights grow*. However, we acknowledge that our experiments compare only against the unregularized dense baseline; a controlled comparison against dropout at equivalent effective sparsity, or against plain Top-K without homeostatic boosting, would be necessary to attribute the gains specifically to the competitive homeostatic mechanism rather than to activation sparsity in general.

5.4 Limitations

1. **Single seed**: All comparisons derive from seed 42; multi-seed replication ($N \geq 3$) is required for variance-aware conclusions.
2. **Single dataset and detector**: PASCAL VOC and YOLOv11n only; generalization to COCO/other architectures is untested.
3. **Training cost**: Sparse training is 2.15× slower—an upfront investment that may be amortized over many deployment instances, though the end-to-end cost-benefit depends on whether sparse hardware delivers the estimated MAC savings.
4. **No sparse hardware validation**: The estimated 48.5% backbone MAC reduction has not been validated on actual sparse accelerators or specialized kernels; measured latency on standard CPU hardware is unchanged.
5. **Single sparsity level**: Only r=0.5 is validated; the full accuracy-efficiency Pareto frontier remains unmapped.
6. **HTM-specific contribution not isolated**: The current experiments do not separate the contribution of homeostatic boosting from plain Top-K competition. Ablating boosting against vanilla Top-K is necessary to confirm the HTM-specific benefit.

6 Conclusion

We demonstrate that HTM-inspired k-Winner activation sparsity, injected into a YOLOv11n backbone, acts as a unified mechanism that simultaneously improves accuracy and introduces activation-level sparsity in this controlled setting. From a single architectural modification:

- Detection accuracy improves by +17.3% relative mAP@0.50 over the dense baseline, with kWTA active at inference—accuracy and activation-level sparsity are co-achieved.
- Robustness to noise, blur, and occlusion improves by +29.6% relative mean corrupted mAP@0.50, with the sparse model favored in all 15 test conditions.

- An estimated 48.5% of backbone MACs become eliminable under sparsity-aware execution, with no accuracy sacrifice required.
- INT8 post-training quantization causes negligible degradation for both models (<0.001 mAP), confirming orthogonal deployability.

These findings suggest that biologically-inspired activation sparsity is a promising mechanism for improving neural network representations in object detection. The co-adaptation between weights and the kWTA operator indicates that competitive activation selection may offer benefits beyond simple regularization, though isolating the specific contribution of homeostatic boosting from generic Top-K competition remains an important direction. Future work will extend to multi-seed validation, MS COCO evaluation, HTM-component ablations (Top-K vs. homeostatic kWTA), graduated sparsity curricula, and deployment on sparse accelerators.

Reproducibility Statement

Code is available at `https://github.com/Mohamed2Elsherbiny/epfl-sparse-cnn-detection`. Random seed 42 is fixed throughout. Training uses Ultralytics YOLOv11n on PASCAL VOC 2007+2012. All evaluation is on VOC 2007 test (4,952 images). The experimental pipeline is fully deterministic given the fixed seed and framework version (Ultralytics 8.4.113, PyTorch 2.x).

Acknowledgments

The author thanks the EPFL Europe AI Summer Research program for funding the AWS SageMaker compute used in the training and evaluation experiments reported in this paper.

References

[1] Subutai Ahmad and Jeff Hawkins. How do neurons operate on sparse distributed representations? A mathematical theory of sparsity, neurons, and active dendrites. *Numenta Technical Report*, 2016.

[2] Subutai Ahmad and Jeff Hawkins. Properties of sparse distributed representations and their application to hierarchical temporal memory. *Numenta Technical Report*, 2016.

[3] Xiaokang Chen, Zehuan Liu, Saining Xie, and Kaiming He. Sparsevit: Revisiting activation sparsity for efficient high-resolution vision transformer. In *Proc. CVPR*, 2023.

[4] Yu Cui, Subutai Ahmad, and Jeff Hawkins. The HTM spatial pooler—a neocortical algorithm for online sparse distributed coding. *Frontiers in Computational Neuroscience*, 11:111, 2017.

[5] Jonathan Frankle and Michael Carbin. The lottery ticket hypothesis: Finding sparse, trainable neural networks. In *Proc. ICLR*, 2019.

[6] Xavier Glorot, Antoine Bordes, and Yoshua Bengio. Deep sparse rectifier neural networks. In *Proc. AISTATS*, 2011.

[7] Marco Grimaldi, Rania Mahmoud, and Matteo Matteucci. Accelerating deep neural networks via semi-structured activation sparsity. In *Proc. ICCV Workshops (RCV)*, 2023.

[8] Song Han, Jeff Pool, John Tran, and William Dally. Learning both weights and connections for efficient neural networks. In *Proc. NeurIPS*, 2015.

[9] Jeff Hawkins and Subutai Ahmad. Why neurons have thousands of synapses, a theory of sequence memory in neocortex. *Frontiers in Neural Circuits*, 10:23, 2016.

[10] Jeff Hawkins and Sandra Blakeslee. *On Intelligence*. Times Books, 2004.

[11] Dan Hendrycks and Thomas Dietterich. Benchmarking neural network robustness to common corruptions and perturbations. In *Proc. ICLR*, 2019.

[12] Andrew G. Howard and et al. Mobilenets: Efficient convolutional neural networks for mobile vision applications. *arXiv:1704.04861*, 2017.

[13] Benoit Jacob and et al. Quantization and training of neural networks for efficient integer-arithmetic-only inference. In *Proc. CVPR*, 2018.

[14] Glenn Jocher and Jing Qiu. Ultralytics yolo11, 2024.

[15] Aditya Kusupati and et al. Soft threshold weight reparameterization for learnable sparsity. In *Proc. ICML*, 2020.

[16] Yunfei Luo, Jia Li, Boxi Peng, Gang Qi, and Yi Yang. Emergence of shape bias in convolutional neural networks through activation sparsity. In *Proc. NeurIPS*, 2023.

[17] Alireza Makhzani and Brendan Frey. k-sparse autoencoders. In *Proc. ICLR*, 2014.

[18] Joseph Redmon, Santosh Divvala, Ross Girshick, and Ali Farhadi. You only look once: Unified, real-time object detection. In *Proc. CVPR*, 2016.

[19] Mark Sandler, Andrew Howard, Menglong Zhu, Andrey Zhmoginov, and Liang-Chieh Chen. Mobilenetv2: Inverted residuals and linear bottlenecks. *Proc. CVPR*, 2018.

[20] Mingxing Tan and Quoc V. Le. Efficientnet: Rethinking model scaling. *Proc. ICML*, 2019.

[21] Xiangyu Zhang, Xinyu Zhou, Mengxiao Lin, and Jian Sun. Shufflenet: An extremely efficient convolutional neural network for mobile devices. *Proc. CVPR*, 2018.

A Algorithm Pseudocode

Algorithm 1 details the k-Winner activation with homeostatic boosting as applied at each injected layer during both training and inference. During training, duty-cycle tracking and boost factors encourage balanced channel utilization; at inference, the boosting is disabled and only the top-r competition is applied.

Algorithm 1 k-Winner Activation with Homeostatic Boosting

Require: $x \in \mathbb{R}^{B \times C \times H \times W}$, retention ratio r, boost strength β
Ensure: $y \in \mathbb{R}^{B \times C \times H \times W}$

$k \leftarrow \max(1, \lfloor r \cdot C \rfloor)$
if training **then**
    $d \leftarrow 0.99 \cdot d + 0.01 \cdot \text{mean}(\mathbf{1}[x > 0], \text{dims} = [B, H, W])$
    $b \leftarrow \exp(-\beta(d/r - 1))$ ▷ Boost under-active channels
    $\tilde{x} \leftarrow b \cdot x$
else
    $\tilde{x} \leftarrow x$
end if
$\tau \leftarrow \text{kth-largest}(\tilde{x}, k, \text{dim} = C)$
$m \leftarrow (\tilde{x} \geq \tau)$ ▷ Binary winner mask
$y \leftarrow x \cdot m$ ▷ Output uses original (unboosted) values
return y

B Robustness Evaluation Protocol

Three corruption types applied to VOC 2007 test images at five severity levels:

- **Gaussian noise**: $\sigma = 10 \times$ severity (additive, per-pixel)
- **Gaussian blur**: kernel radius = severity pixels
- **Occlusion**: gray rectangle covering $10\% \times$ severity of image area

No models are retrained; evaluation uses frozen checkpoints. Statistics are paired across 15 severity-point cells of a single checkpoint pair.

C Training Convergence

Table 6: Validation mAP@0.50:0.95 at selected epochs and total training time.

Epoch	5	10	15	20	25	30
Baseline	0.070	0.180	0.275	0.337	0.380	0.394
HTM r=0.5	0.081	0.195	0.271	0.317	0.361	0.370

The sparse model converges 6.2% below the baseline on the validation set (where kWTA is active during evaluation) but surpasses it by 25.6% on the test set when kWTA remains active at inference (Mode C). This gap between validation and test performance warrants further investigation in future work.

Figures

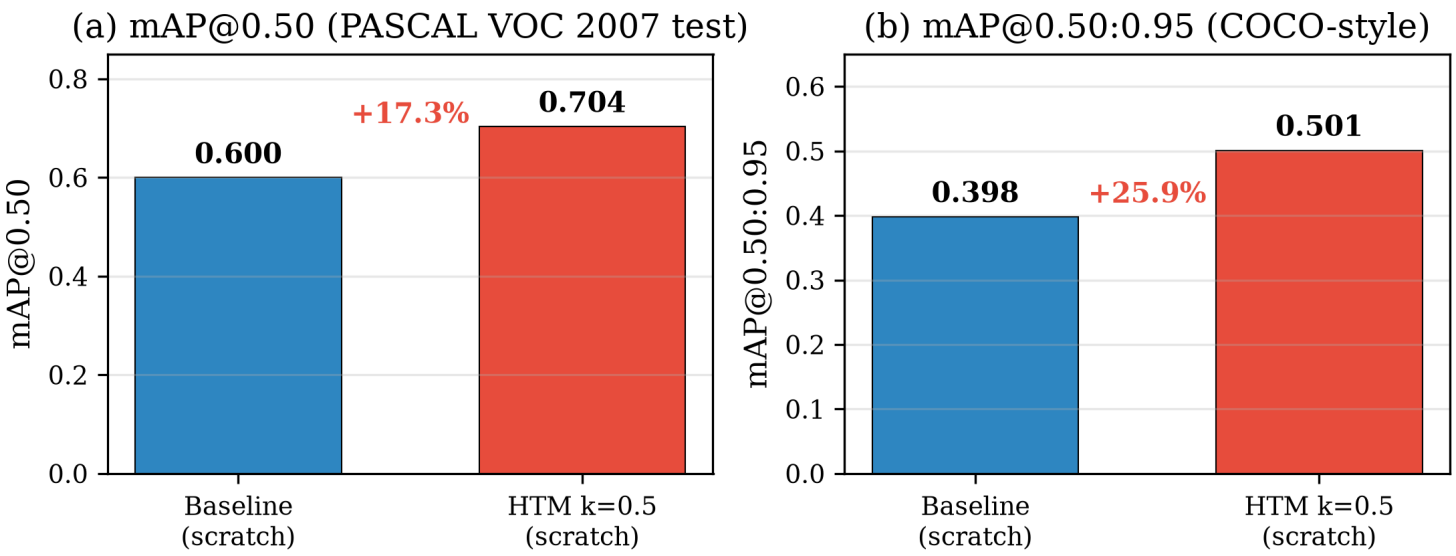


Figure 1

Detection performance on VOC 2007 test set (dense inference). (a) mAP@0.50. (b) mAP@0.50:0.95. HTM k=0.5 outperforms the dense baseline by +17.3% and +25.9% relative, respectively.

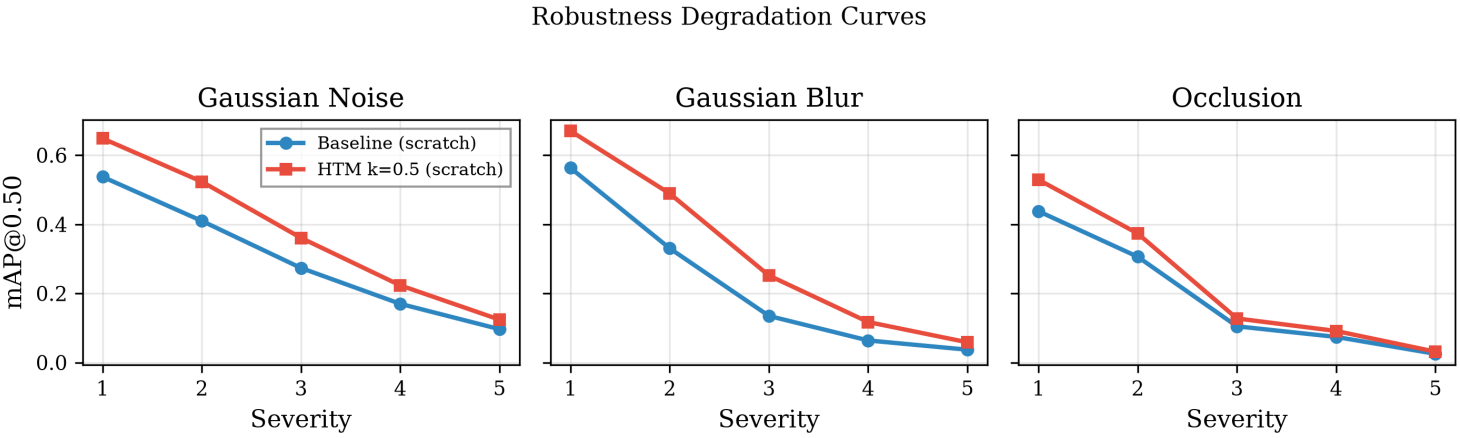


Figure 2

Robustness degradation curves across severity levels 1–5. HTM k=0.5 (red) degrades more gracefully than baseline (blue) under all three corruption types.

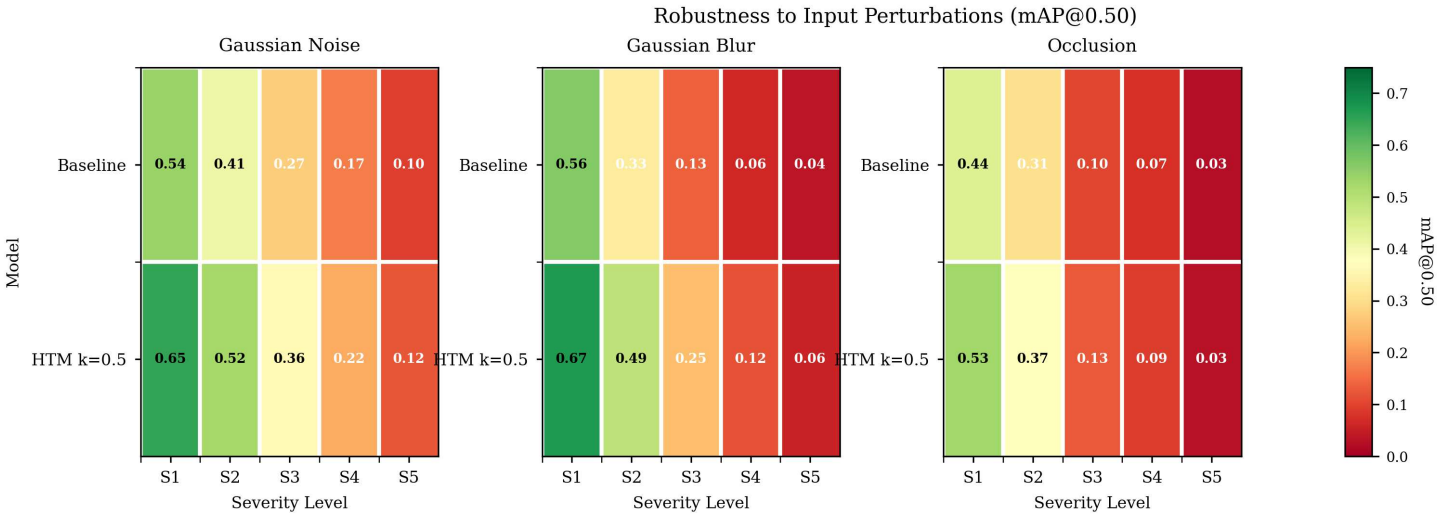


Figure 3

Robustness heatmap: mAP@0.50 across 3 corruptions × 5 severities. HTM k=0.5 (bottom row) outperforms baseline (top row) at every severity level.