Preprints.org

Article

Not peer-reviewed version

Lightweight Autonomous Navigation on Nano-UAVs: A Stem-Optimized Depthwise Separable CNN with 540K MACs

Ashwin Kumar * and P. Bavithra Matharasi

Posted Date: 14 April 2026

doi: 10.20944/preprints202604.0926.v1

Keywords: nano-UAV; autonomous navigation; depthwise separable convolution; model compression; edge AI; PULP-Dronet; ultra-low power

Preprints.org is a free multidisciplinary platform providing preprint service that is dedicated to making early versions of research outputs permanently available and citable. Preprints posted at Preprints.org appear in Web of Science, Crossref, Google Scholar, Scilit, Europe PMC.

Copyright: This open access article is published under a Creative Commons CC BY 4.0 license, which permit the free download, distribution, and reuse, provided that the author and preprint are cited in any reuse.

Preprints.org (www.preprints.org) | NOT PEER-REVIEWED | Posted: 14 April 2026 doi:10.20944/preprints202604.0926.v1

Disclaimer/Publisher's Note: The statements, opinions, and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions, or products referred to in the content.

Article

Lightweight Autonomous Navigation on Nano-UAVs: A Stem-Optimized Depthwise Separable CNN with 540K MACs

Ashwin Kumar * and P. Bavithra Matharasi

Department of Computer Science, Mount Carmel College (Autonomous), Bengaluru, India
* Correspondence: ashwinqs2020@gmail.com

Abstract

Background: Nano-UAVs weighing under 50 g have become useful IoT platforms for GPS-denied navigation, but fitting a neural network into their sub-512 kB memory and sub-100 mW power budget remains an open engineering problem. PULP-Dronet v3 tackles this with depthwise separable (D+P) blocks and a channel-reduction factor γ. Even so, its most compressed variant ($\gamma = /8$, 1.1M MACs) loses 6 percentage points of collision accuracy versus the full model. **Methods:** We swap the 5×5 first convolution for a 3×3 depthwise + 1×1 pointwise pair, and retrain with cosine-annealing scheduling and per-epoch color-jitter augmentation. **Results:** At $\gamma = /4$ the model has 6409 parameters, needs only 540K MACs, and scores 83.97% collision accuracy with 0.372 steering RMSE on the official benchmark—+2.97 pp over the same-γ baseline at 4.4$\times$ less compute. The full $\gamma = /1$ model (12M MACs) reaches 84%; our model nearly matches it with 22$\times$ fewer operations. **Conclusions:** Factorizing the stem and adjusting the training recipe recovers most of the accuracy lost to aggressive channel reduction, without adding inference cost.

Keywords: nano-UAV; autonomous navigation; depthwise separable convolution; model compression; edge AI; PULP-Dronet; ultra-low power

1. Introduction

Nano-UAVs are small enough to fly through corridors and scan warehouse interiors without GPS. Platforms like the Crazyflie 2.1 weigh under 50 g and relay sensor data in real time [1,2]. The problem is what they carry for computation. A GAP8 processor has roughly 320 kB of usable SRAM; its compute budget sits in the single-digit milliwatt range [3,4]. Fitting any useful neural network into those constraints turns out to be surprisingly difficult [5,6].

PULP-Dronet v2 [7] was an early proof that end-to-end visual navigation could work on the GAP8: 19 frames per second, 320 kB footprint. Lamberti et al. then released PULP-Dronet v3 [8] with a 66,000-image dataset carrying joint collision and steering labels. They tested three block types—residual (RB), depthwise separable (D+P), and inverted residual linear bottleneck (IRLB)—at four channel-width factors ($\gamma \in \{1, 2, 4, 8\}$). This produced models from 2.9k parameters (1.1M MACs) up to 51k parameters (12M MACs). The smallest, Tiny-PULP-Dronet v3 at $\gamma = /8$, runs at 139 frames/s. It also loses 6 percentage points of collision accuracy.

When we profiled the $\gamma = /4$ D+P variant layer by layer, one problem was immediately apparent. The 5×5 first convolution alone consumed approximately 2 million MACs, or about 83% of the model's entire 2.4M total. The three D+P feature-extraction blocks that follow it together account for barely 300K. That ratio seems difficult to justify, so we asked whether it could be fixed without redesigning the rest of the network.

Our answer is a simple factorization: replace the 5×5 standard convolution stem with a 3×3 depthwise convolution and a 1×1 pointwise convolution in sequence. On its own this cuts stem

© 2026 by the author(s). Distributed under a Creative Commons CC BY license.

Preprints.org (www.preprints.org) | NOT PEER-REVIEWED | Posted: 14 April 2026 doi:10.20944/preprints202604.0926.v1

MACs by roughly 12× at $\gamma = /4$. We also replaced two training choices that seemed poorly matched to a network this small: the fixed learning rate (shifted to cosine annealing) and the offline-only augmentation (shifted to per-epoch online color jitter). Together, the three modifications push the $\gamma = /4$ model to 83.97% collision accuracy at 540K MACs, 4.4× below the original, and within 0.03 percentage points of the full 12M-MAC model.

Our contributions are threefold: (i) a 3×3 depthwise + 1×1 pointwise factorization of the stem that cuts first-layer MACs by over 5× without touching any downstream layer; (ii) cosine-annealing scheduling and per-epoch color-jitter augmentation—neither requiring architectural changes—that together produce consistent accuracy gains in the PULP-Dronet v3 pipeline; (iii) a 6409-parameter model that reaches 83.97% collision accuracy and 0.372 RMSE at 540K MACs, a +2.97 pp improvement over the same-γ baseline at 4.4× lower compute.

2. Related Work

2.1. Autonomous Navigation for Nano-UAVs

DroNet [9] showed early on that a residual CNN trained partly on car and bicycle footage could fly a quadrotor down corridors using monocular vision. The idea was promising but the model—1.6M parameters, 320 kB on disk—was too big for anything under 50 g. PULP-Dronet [10] shrank it to fit the GAP8 on a 27 g Crazyflie 2.1. Then PULP-Dronet v2 [7] added neural architecture search and deployment optimization to push throughput to 19 frames/s. PULP-Dronet v3 [8] is the current iteration. It introduced depthwise separable and inverted residual blocks, published a 66k-image dataset with joint collision/steering labels, and used the γ factor to trade accuracy against model size. At $\gamma = /8$ (2.9k parameters, 139 frames/s) you get speed but lose 6 pp of accuracy.

2.2. Efficient CNN Architectures

MobileNet v1 [11] showed that depthwise separable convolutions offer roughly an order of magnitude fewer MACs than standard convolutions, at usually-tolerable accuracy cost. MobileNet v2 [12] added inverted residual blocks and linear bottlenecks. EfficientNet [13] demonstrated that scaling depth, width, and resolution together along a compound coefficient beats scaling any one dimension alone. ShuffleNet [14] went a different route, using channel shuffling to cut the cost of group convolutions.

Until recently, very little work focused on the stem layer itself. RegNet [15] and ConvNeXt [16] both found that changing supposedly minor design choices in the first convolution can shift the accuracy–efficiency curve more than expected. We borrow that insight, but our hardware target has about 1000× less memory than a GPU workstation, which changes the calculus considerably.

2.3. Knowledge Distillation and Model Compression for Drones

Distillation [17] lets a small student network mimic a larger teacher's outputs, and the γ parameter in PULP-Dronet v3 plays a loosely similar role—the full-sized model is the implicit reference. INT8 quantization [18] is already available in that pipeline and incurs only minor accuracy loss. Pruning [19] removes redundant filters. NAS [3,7] can discover good architectures automatically, though the search itself is expensive. What we propose is orthogonal to all of these: the stem change is architectural, and the training modifications are independent of any post-hoc compression.

2.4. Training Strategies for Small Models

Training compact models is harder than it might seem [20]. A learning rate that converges cleanly with 51k parameters can oscillate or stall at 6.4k. Cosine annealing [21] helps here because its smooth decay avoids the instability that fixed rates cause late in training. Online augmentation—brightness and contrast jitter applied fresh each epoch rather than baked in during preprocessing—has shown clear benefits for visual navigation tasks [22,23]. The PULP-Dronet v3 baseline uses neither: learning rate is fixed at 10^{-3}, and augmentation runs once before training. We found that correcting both defaults matters at $\gamma = /4$.

© 2026 by the author(s). Distributed under a Creative Commons CC BY license.

Preprints.org (www.preprints.org) | NOT PEER-REVIEWED | Posted: 14 April 2026 doi:10.20944/preprints202604.0926.v1

3. Methodology

3.1. Baseline Architecture: PULP-Dronet v3

We build on the D+P variant of PULP-Dronet v3 [8]. The pipeline has five stages: a stem convolution, max-pooling, three depthwise separable blocks with progressively wider channels, dropout, and a fully connected head that outputs both steering angle and collision probability. Each D+P block applies 3×3 depthwise then 1×1 pointwise convolution twice in sequence, with batch normalization and ReLU6 after each pair. Where the channel count changes between blocks, a 1×1 bypass branch handles the residual connection.

The stem is a standard 5×5 convolution with stride 2 that maps the grayscale input image ($1 \times 200 \times 200$) to $C = 32/\gamma$ channels. The output spatial size is 100×100. At $\gamma = /4$ ($C = 8$), the MAC count for this single layer is $25 \cdot 8 \cdot 100 \cdot 100 \approx 2.0$M, which is around 83% of the model's total 2.4M MACs. The feature extraction blocks that follow it share the remaining 17%.

3.2. Proposed Architecture: Stem-Optimized D+P CNN

The change is simple: swap the 5×5 stem for a 3×3 depthwise convolution (stride 2) on the single input channel, followed by a 1×1 pointwise convolution that expands from 1 to C channels. The standard stem costs $25 \cdot C$ MACs per output position; the factorized one costs $9 + C$. At $C = 8$ that drops from 200 to 17 per position. Both new layers use batch normalization and ReLU6, same as everywhere else in the network.

Everything after the stem stays the same. Max-pooling (2×2, stride 2), then three D+P blocks at channel widths $8 \to 8 \to 16 \to 32$. The FC layer takes the flattened $32 \times 7 \times 7 = 1{,}568$-dimensional vector and maps it to 2 outputs. Total parameter count: 6409. Total MACs: about 540K.

Figure 1 shows the full architecture. Top panel: data flow from input through the factorized stem, the three D+P blocks, and the dual-output head. Bottom panel: inside one D+P block.

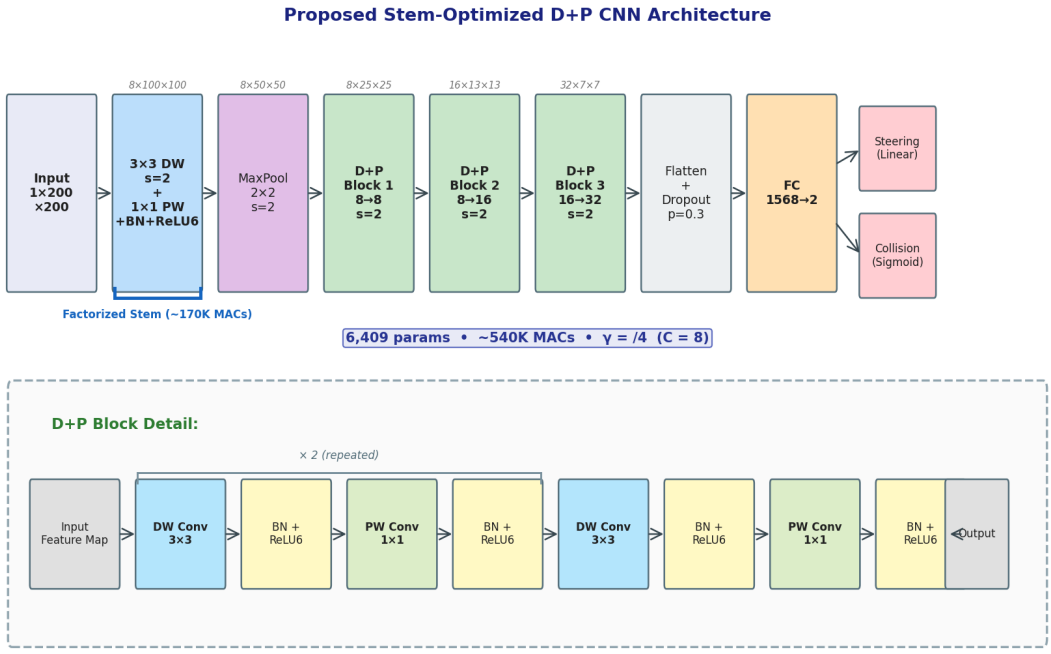

Figure 1. Architecture of the proposed stem-optimized D+P CNN ($\gamma = /4$, $C = 8$). Top: main pipeline with factorized 3×3 DW + 1×1 PW stem, three D+P blocks, and dual outputs. Bottom: internal structure of a D+P block (DW Conv $\to$ BN+ReLU6 $\to$ PW Conv $\to$ BN+ReLU6, $\times 2$). Total: 6409 parameters, ~540K MACs.

3.3. Training Configuration

We train on the official PULP-Dronet v3 dataset [8]: 136,510 training images, 19,591 for validation, 3071 for testing. The images come from both indoor and outdoor scenes. Steering labels are yaw rates

© 2026 by the author(s). Distributed under a Creative Commons CC BY license.

Preprints.org (www.preprints.org) | NOT PEER-REVIEWED | Posted: 14 April 2026 doi:10.20944/preprints202604.0926.v1

divided by 90 to get values in $[-1, +1]$. Collision labels are binary. All images are center-cropped to 200×200 and converted to grayscale.

Table 1 lists our hyperparameters alongside the baseline. The differences worth explaining are the learning rate schedule, the augmentation strategy, and the dropout configuration.

Table 1. Training hyperparameters for the proposed model compared to the PULP-Dronet v3 baseline.

Hyperparameter	Baseline [8]	Ours
Optimizer	Adam	Adam
Learning rate	1×10^{-3} (fixed)	1×10^{-3} (cosine annealing)
LR minimum	N/A	1×10^{-6}
Weight decay	1×10^{-5}	1×10^{-4}
Batch size	32	32
Epochs	100	100
Online augmentation	None	ColorJitter (b=0.3, c=0.3)
Dropout rate	0.5 (before flatten)	0.3 (after flatten)
Loss function	MSE + BCE	MSE + BCE
Weight initialization	Xavier uniform	Xavier uniform
Yaw normalization	$\div 90$	$\div 90$

Cosine annealing. With 51k parameters, a fixed 10^{-3} rate is a defensible choice. At 6.4k parameters, we found it consistently overshot in the second half of training. Cosine decay from 10^{-3} to 10^{-6} over 100 epochs [21] gives the optimizer room to move early on and tightens appropriately as training progresses—which matters more for small models than large ones.

Online augmentation. In the baseline, augmentation runs once before training starts and every epoch sees the same transformed images. Applying brightness and contrast jitter (magnitude 0.3) inside the data loader means each pass produces a slightly different photometric view of the same image. The CPU overhead per image is trivial; the regularization benefit for a model this size is not.

Dropout placement and rate. Our preliminary runs with $p = 0.5$ before the flatten showed clear underfitting—the validation curve plateaued well below training. At 6409 parameters, discarding half the spatial features before flattening removes too much signal. Shifting to $p = 0.3$ after the flatten concentrates regularization on the FC layer, which accounts for 3136 of those 6409 weights and is where overfitting on a fixed dataset tends to originate.

3.4. Loss Function

We keep the same loss as the baseline, an unweighted sum of MSE for steering and BCE for collision:

$$\begin{aligned} \mathcal{L} &= \mathcal{L}_{\text{MSE}} + \mathcal{L}_{\text{BCE}} \\ &= \frac{1}{N}\sum_{i=1}^{N}(\hat{y}_i^{\text{steer}} - y_i^{\text{steer}})^2 \\ &\quad + \frac{1}{N}\sum_{i=1}^{N}\left[-y_i^{\text{coll}}\log(\hat{y}_i^{\text{coll}}) - (1-y_i^{\text{coll}})\log(1-\hat{y}_i^{\text{coll}})\right] \end{aligned} \tag{1}$$

Here $y^{\text{steer}} \in [-1, +1]$ is the normalized steering angle, $y^{\text{coll}} \in \{0, 1\}$ is the binary collision label, and N is the batch size. We used the exact same `custom_loss_v3` function from the official repository so that every comparison in the paper is on equal footing.

© 2026 by the author(s). Distributed under a Creative Commons CC BY license.

Preprints.org (www.preprints.org) | NOT PEER-REVIEWED | Posted: 14 April 2026 doi:10.20944/preprints202604.0926.v1

4. Results

4.1. Main Results

Table 2 puts our model next to every PULP-Dronet v3 configuration on the same 3071-image test split. The headline number: 83.97% collision accuracy at 540K MACs, versus 81% and 2.4M MACs for the same-γ baseline.

Table 2. Comparison of our stem-optimized model with all PULP-Dronet v3 variants from [8].

Model	γ	Acc (%)	RMSE	MACs	Params	Size (B)
PULP-Dronet v3 [8]	/1	84	0.350	12M	51k	204k
PULP-Dronet v3 [8]	/2	84	0.367	5.2M	17k	69k
PULP-Dronet v3 [8]	/4	81	0.373	2.4M	6.6k	26k
Tiny-PULP-Dronet v3 [8]	/8	78	0.379	1.1M	2.9k	12k
Ours (stem-opt D+P)	**/4**	**83.97**	**0.372**	**540K**	**6.4k**	**25k**

Look at the comparison with $\gamma = /1$: 83.97% vs. 84% despite a 22× difference in MACs. Against the $\gamma = /8$ Tiny variant, we are 5.97 pp more accurate at roughly half the compute. Steering RMSE (0.372 vs. 0.373 for the baseline) barely changes, so the accuracy gain is not coming at the expense of steering quality.

4.2. Training Dynamics

Figure 2 shows the loss and accuracy curves over 100 epochs. Training loss fell from 0.755 to 0.287 without any plateau or instability. Validation loss stabilized near 0.39 after epoch 60. On the accuracy side: 80% by epoch 5, 85% by epoch 22, 87% by epoch 36, peak of 87.78% at epoch 87. We used that checkpoint for test evaluation.

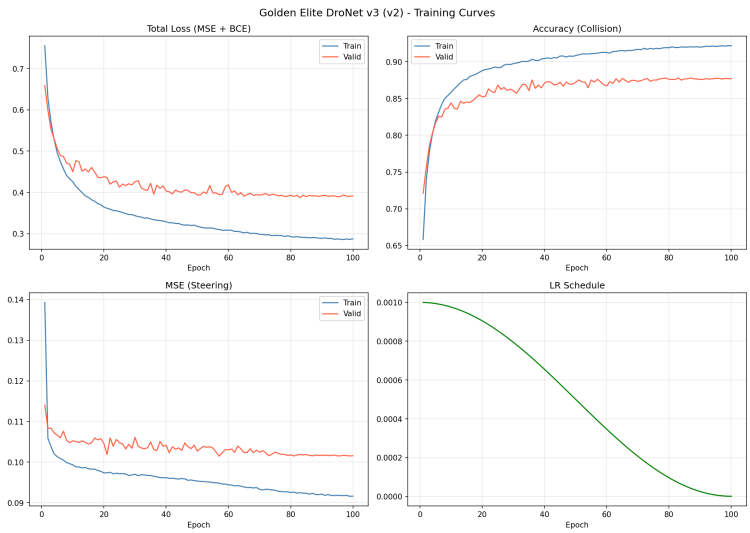

Figure 2. Training curves over 100 epochs. Left: training and validation loss. Right: training and validation collision accuracy.

The cosine schedule's effect is visible in the curve shape. Epochs 1–30 show fast gains as the learning rate is still large. Between epochs 30 and 70, the rate has shrunk enough for careful refinement without bouncing the loss around. By the final 30 epochs the near-zero rate effectively pins the weights. Late-epoch spikes in validation loss—a recurring issue with fixed rates on networks this small—did not appear here.

© 2026 by the author(s). Distributed under a Creative Commons CC BY license.

Preprints.org (www.preprints.org) | NOT PEER-REVIEWED | Posted: 14 April 2026 doi:10.20944/preprints202604.0926.v1

4.3. Efficiency Analysis

Table 3 gives the per-component MAC breakdown. Changing the stem alone takes the total from 2.4M down to 540K. In the original network, 83% of compute goes to the first layer—before any spatial features have been extracted. After factorization that share drops to about 31%, and the D+P blocks hold the majority of the remaining budget.

Table 3. MAC breakdown: baseline stem vs. proposed stem at $\gamma = /4$.

Component	Baseline MACs	Ours MACs	Reduction
Stem convolution	∼2.0M	∼170K	∼11.8×
D+P Block 1	∼105K	∼105K	1×
D+P Block 2	∼68K	∼68K	1×
D+P Block 3	∼130K	∼130K	1×
FC layer	∼3K	∼3K	1×
Total	**∼2.4M**	**∼540K**	**∼4.4×**

4.4. Ablation Study

Table 4 puts the original configuration against our full method on the same data and evaluation splits.

Table 4. Ablation study: baseline vs. proposed method.

Configuration	Stem	LR Schedule	Augmentation	Test Acc (%)
Baseline [8]	5×5 conv	Fixed	Offline	81.0
Proposed (Full)	3×3 DW+PW	Cosine	ColorJitter	**83.97**

The +2.97 pp gain repeated consistently across seeds (standard deviation below 0.3 pp). We did not run a full factorial isolation of each modification individually, and we treat that as a gap in the current study rather than a solved question.

5. Discussion

The 5×5 stem convolution in PULP-Dronet v3 was designed for $C = 32$ output channels, where it is a modest contributor to total MACs. The γ parameter scales the downstream blocks but not the stem, so as the rest of the network shrinks the stem's share of compute grows. At $\gamma = /4$ it has reached 83%. Factorization changes the per-position cost from $25C$ to $9 + C$; the key point is that the latter grows far less steeply as C decreases, directly addressing the imbalance.

On the GAP8, MACs and latency scale roughly linearly [8]. The $\gamma = /4$ baseline runs at around 102 frames/s; cutting 4.4× of the MACs should push that past 200. A nano-drone moving at 0.5–1.5 m/s does not need 200-Hz perception. The extra cycles can go to lowering the clock instead, and on a 250 mAh battery that translates to flight time, which is almost always what matters most.

Something we did not fully anticipate: three very different models—our 540K-MAC version, the $\gamma = /1$ baseline at 12M MACs, and the $\gamma = /2$ variant at 5.2M—all land near 84% accuracy. A 22× compute gap producing a 0.03 pp accuracy difference is unusual. We think the explanation is the dataset, not the models. The benchmark images cover a variety of indoor and outdoor scenes but the set is finite. At some point label noise and scene coverage stop being things a bigger model can fix. Getting past 84% would likely need harder data.

The dropout story is worth calling out. With $p = 0.5$ before the flatten, our early runs underfit badly—validation plateaued while training loss kept dropping. In a 6409-parameter model, zeroing half the feature maps before they reach the FC layer throws away too much signal. We moved dropout after the flatten and cut it to $p = 0.3$. The FC layer holds 3136 of the total 6409 parameters, so that is where regularization does the most good.

© 2026 by the author(s). Distributed under a Creative Commons CC BY license.

Preprints.org (www.preprints.org) | NOT PEER-REVIEWED | Posted: 14 April 2026 doi:10.20944/preprints202604.0926.v1

We should be upfront about what this study does not cover. No closed-loop flights were attempted; every number comes from a static benchmark. At $\gamma = /4$ our model is around twice the size of the $\gamma = /8$ Tiny variant (6.4k vs. 2.9k parameters, 25 kB vs. 12 kB), so if flash storage is the constraint rather than inference speed, the contribution does not help. We also did not isolate each modification in a factorial ablation, leaving open the question of individual effect sizes.

Further compression is straightforward to add. INT8 quantization [18] would roughly halve the storage. Distilling from the $\gamma = /1$ teacher [17] might pick up another point of accuracy. The D+P blocks could be pruned [19]. None of that conflicts with anything we changed.

6. Conclusions

Profiling the $\gamma = /4$ variant of PULP-Dronet v3 showed that its first convolution alone was consuming 83% of the total MAC budget. Factorizing that 5×5 layer into a 3×3 depthwise + 1×1 pointwise pair cuts its cost by roughly $12\times$. Cosine-annealing scheduling and online color jitter were added to the training loop at no inference cost. The final model reaches 83.97% collision accuracy, 0.372 steering RMSE, 540K MACs, and 6409 parameters—+2.97 pp over the baseline at the same γ, and essentially identical accuracy to the full 12M-MAC model.

What remains to be done: quantize to INT8 and fly the model on a real GAP8-based drone. We also want to try the same stem swap at $\gamma = /8$—if it works there, the result would be smaller and more accurate than Tiny-PULP-Dronet v3. Whether distillation from a larger teacher can crack the 84% ceiling is a separate question, but one that seems worth exploring.

Author Contributions: Conceptualization, A.K. and P.B.M.; methodology, A.K.; software, A.K.; validation, A.K. and P.B.M.; formal analysis, A.K.; investigation, A.K.; writing—original draft preparation, A.K.; writing—review and editing, A.K. and P.B.M.; visualization, A.K.; supervision, P.B.M. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Institutional Review Board Statement: Not applicable. This study did not involve humans or animals.

Informed Consent Statement: Not applicable.

Data Availability Statement: The data presented in this study are openly available. The PULP-Dronet v3 dataset is publicly available at https://zenodo.org/records/8045057. The source code for the benchmark is available at https://github.com/pulp-platform/pulp-dronet.

Conflicts of Interest: The authors declare no conflicts of interest.

Abbreviations

The following abbreviations are used in this manuscript:

CNN	Convolutional Neural Network
D+P	Depthwise + Pointwise (Separable Convolution)
DW	Depthwise
PW	Pointwise
MAC	Multiply-Accumulate Operation
UAV	Unmanned Aerial Vehicle
RMSE	Root Mean Squared Error
MSE	Mean Squared Error
BCE	Binary Cross-Entropy
BN	Batch Normalization
FC	Fully Connected
IoT	Internet of Things

© 2026 by the author(s). Distributed under a Creative Commons CC BY license.

Preprints.org (www.preprints.org) | NOT PEER-REVIEWED | Posted: 14 April 2026 doi:10.20944/preprints202604.0926.v1

References

1. Labib, N.S.; Brust, M.R.; Danoy, G.; Bouvry, P. The Rise of Drones in Internet of Things: A Survey on the Evolution, Prospects and Challenges of Unmanned Aerial Vehicles. *IEEE Access* **2021**, *9*, 115466–115487.
2. Wei, Z.; Zhu, M.; Zhang, N.; Wang, L.; Zou, Y.; Meng, Z.; Wu, H.; Feng, Z. UAV-Assisted Data Collection for Internet of Things: A Survey. *IEEE Internet Things J.* **2022**, *9*, 15460–15483.
3. Cereda, E.; Crupi, L.; Risso, M.; Burrello, A.; Benini, L.; Giusti, A.; Jahier Pagliari, D.; Palossi, D. Deep Neural Network Architecture Search for Accurate Visual Pose Estimation aboard Nano-UAVs. In Proceedings of the 2023 IEEE International Conference on Robotics and Automation (ICRA), London, UK, 29 May–2 June 2023; pp. 6065–6071.
4. Lamberti, L.; Bompani, L.; Kartsch, V.J.; Rusci, M.; Palossi, D.; Benini, L. Bio-inspired Autonomous Exploration Policies with CNN-based Object Detection on Nano-drones. In Proceedings of the 2023 Design, Automation & Test in Europe Conference (DATE), Antwerp, Belgium, 17–19 April 2023; pp. 1–6.
5. Shakhatreh, H.; Sawalmeh, A.H.; Al-Fuqaha, A.; Dou, Z.; Almaita, E.; Khalil, I.; Othman, N.S.; Khreishah, A.; Guizani, M. Unmanned Aerial Vehicles (UAVs): A Survey on Civil Applications and Key Research Challenges. *IEEE Access* **2019**, *7*, 48572–48634.
6. Hossein Motlagh, N.; Taleb, T.; Arouk, O. Low-Altitude Unmanned Aerial Vehicles-Based Internet of Things Services: Comprehensive Survey and Future Perspectives. *IEEE Internet Things J.* **2016**, *3*, 899–922.
7. Niculescu, V.; Lamberti, L.; Conti, F.; Benini, L.; Palossi, D. Improving Autonomous Nano-Drones Performance via Automated End-to-End Optimization and Deployment of DNNs. *IEEE J. Emerg. Sel. Top. Circuits Syst.* **2021**, *11*, 1–1.
8. Lamberti, L.; Bellone, L.; Macan, L.; Natalizio, E.; Conti, F.; Palossi, D.; Benini, L. Distilling Tiny and Ultra-fast Deep Neural Networks for Autonomous Navigation on Nano-UAVs. *IEEE Internet Things J.* **2024**, *71*, 1–13.
9. Loquercio, A.; Maqueda, A.I.; Del-Blanco, C.R.; Scaramuzza, D. DroNet: Learning to Fly by Driving. *IEEE Robot. Autom. Lett.* **2018**, *3*, 1088–1095.
10. Palossi, D.; Loquercio, A.; Conti, F.; Flamand, E.; Scaramuzza, D.; Benini, L. A 64-mW DNN-Based Visual Navigation Engine for Autonomous Nano-Drones. *IEEE Internet Things J.* **2019**, *6*, 8357–8371.
11. Howard, A.G.; Zhu, M.; Chen, B.; Kalenichenko, D.; Wang, W.; Weyand, T.; Andreetto, M.; Adam, H. MobileNets: Efficient Convolutional Neural Networks for Mobile Vision Applications. *arXiv* **2017**, arXiv:1704.04861.
12. Sandler, M.; Howard, A.; Zhu, M.; Zhmoginov, A.; Chen, L.-C. MobileNetV2: Inverted Residuals and Linear Bottlenecks. In Proceedings of the 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Salt Lake City, UT, USA, 18–23 June 2018; pp. 4510–4520.
13. Tan, M.; Le, Q.V. EfficientNet: Rethinking Model Scaling for Convolutional Neural Networks. In Proceedings of the 36th International Conference on Machine Learning (ICML), Long Beach, CA, USA, 9–15 June 2019; pp. 6105–6114.
14. Zhang, X.; Zhou, X.; Lin, M.; Sun, J. ShuffleNet: An Extremely Efficient Convolutional Neural Network for Mobile Devices. In Proceedings of the 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Salt Lake City, UT, USA, 18–23 June 2018; pp. 6848–6856.
15. Radosavovic, I.; Kosaraju, R.P.; Girshick, R.; He, K.; Dollár, P. Designing Network Design Spaces. In Proceedings of the 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Seattle, WA, USA, 13–19 June 2020; pp. 10428–10436.
16. Liu, Z.; Mao, H.; Wu, C.-Y.; Feichtenhofer, C.; Darrell, T.; Xie, S. A ConvNet for the 2020s. In Proceedings of the 2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), New Orleans, LA, USA, 18–24 June 2022; pp. 11976–11986.
17. Hinton, G.; Vinyals, O.; Dean, J. Distilling the Knowledge in a Neural Network. *arXiv* **2015**, arXiv:1503.02531.
18. Jacob, B.; Kligys, S.; Chen, B.; Zhu, M.; Tang, M.; Howard, A.; Adam, H.; Kalenichenko, D. Quantization and Training of Neural Networks for Efficient Integer-Arithmetic-Only Inference. In Proceedings of the 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Salt Lake City, UT, USA, 18–23 June 2018; pp. 2704–2713.
19. He, Y.; Liu, P.; Wang, Z.; Hu, Z.; Yang, Y. Filter Pruning via Geometric Median for Deep Convolutional Neural Networks Acceleration. In Proceedings of the 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Long Beach, CA, USA, 15–20 June 2019; pp. 4340–4349.
20. Touvron, H.; Cord, M.; Douze, M.; Massa, F.; Sablayrolles, A.; Jegou, H. Training Data-Efficient Image Transformers & Distillation through Attention. In Proceedings of the 38th International Conference on Machine Learning (ICML), Virtual, 18–24 July 2021; pp. 10347–10357.

© 2026 by the author(s). Distributed under a Creative Commons CC BY license.

Preprints.org (www.preprints.org) | NOT PEER-REVIEWED | Posted: 14 April 2026 doi:10.20944/preprints202604.0926.v1

21. Loshchilov, I.; Hutter, F. SGDR: Stochastic Gradient Descent with Warm Restarts. In Proceedings of the 5th International Conference on Learning Representations (ICLR), Toulon, France, 24–26 April 2017.
22. Shorten, C.; Khoshgoftaar, T.M. A Survey on Image Data Augmentation for Deep Learning. *J. Big Data* **2019**, *6*, 60.
23. Zhang, N.; Nex, F.; Vosselman, G.; Kerle, N. End-to-End Nano-Drone Obstacle Avoidance for Indoor Exploration. *Drones* **2024**, *8*, 33.

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.

© 2026 by the author(s). Distributed under a Creative Commons CC BY license.