

DAC-LOOP: DISTILLATION-AWARE COMPRESSION FOR REMOTE SENSING CNN

Rico Staedeli  Lorena Raichle  Konstantinos Tzevelekakis  Aron Asefaw  Damian Borth 

AIML Lab, School of Computer Science, University of St. Gallen

Abstract—While large-scale models achieve strong results in remote sensing scene classification, deploying them on edge devices remains difficult due to limited compute, memory, and energy budgets. Compact CNN backbones are therefore a practical choice in many real-world settings, offering efficient inference with well-optimized implementations on embedded hardware. To push this efficiency further, we investigate how far a knowledge-distilled student model can be compressed while preserving high accuracy. We propose a **Distillation-Aware Compression Loop (DAC-Loop)** that integrates (i) *knowledge distillation* applied both *before and after* structured channel pruning, and (ii) an *iterative compression schedule* that alternates structured pruning with unstructured sparsification. Across EuroSAT and NWPU-RESISC45, the DAC-Loop consistently improves the accuracy-compression trade-off relative to standard one-shot pruning and single-stage KD recovery, maintaining strong accuracy even beyond 90% parameter reduction. These results demonstrate that combining distillation and pruning in a unified iterative loop offers an effective path toward compact remote sensing models. The Code is available in [GitHub](#).¹

Index Terms—remote sensing, model compression, knowledge distillation, pruning, teacher-student learning

I. INTRODUCTION

Remote Sensing (RS) has benefited greatly from advances in deep neural networks, enabling highly accurate land-use and land-cover (LULC) classification on large-scale datasets such as EuroSAT and NWPU-RESISC45 [1]. LULC-classification became a central task for monitoring, environmental analysis, and constrained devices decision-making. However, deploying such models on resource-constrained devices remains challenging as state-of-the-art CNN-backbones require substantial memory, compute, and power [2]. This challenge motivates the development of compact yet accurate RS classifiers. While lightweight architectures exist, they rely on custom, task-specific designs and extensive architecture engineering [3]. A promising direction to mitigate this gap lies in teacher-student knowledge distillation (KD), where a high-capacity teacher model guides a smaller student model in training to achieve comparable accuracy [4] [5] [6]. Yet, even smaller models such as distilled students can exceed the tight memory and compute limits of resource-constrained platforms [7]. The goal of this paper is to explore how much a knowledge-distilled student model can be further compressed while preserving high level of accuracy on RS scene classification. We therefore consider compression as an iterative process in combination

with structured and unstructured pruning. In particular, our main contributions are:

- We introduce the DAC-Loop (Fig. 1), an iterative distillation-aware compression protocol for CNN backbones that alternates pruning and KD-based recovery under a fixed teacher.
- We analyze distillation around structured pruning (KD-1/KD-2) and a pruning schedule that alternates structured and unstructured pruning across iterations.
- On EuroSAT and NWPU-RESISC45, ablations and accuracy-compression curves show that the DAC-Loop improves the accuracy-efficiency trade-off over standard one-shot pruning (Fig. 2).

The DAC-Loop contributes a *compression protocol* which specifies where distillation is applied, how compression is staged across multiple cycles, and how structured and unstructured pruning are combined.

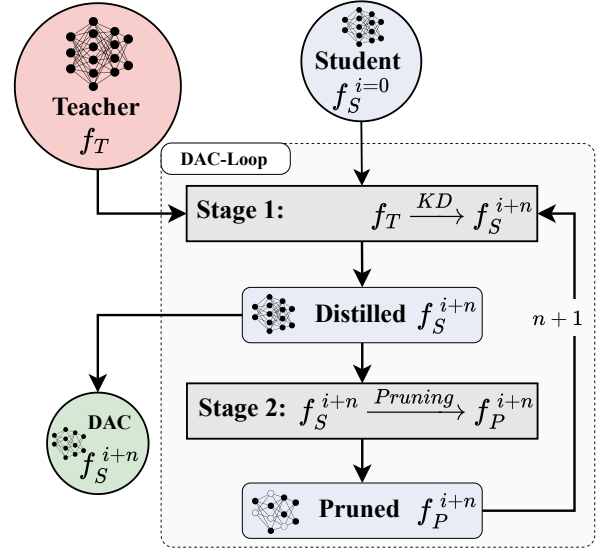


Fig. 1. Distillation-Aware Compression Loop (DAC-Loop). A fixed, pretrained *teacher* network $f_T(\cdot; \theta_T)$ provides supervision for iterative compression of a smaller *student*. Starting from a randomly initialized student $f_S^{i=0}$, *Stage 1* performs KD from f_T to obtain a distilled student f_S^{i+n} . The resulting model is then compressed in *Stage 2* via structured or unstructured pruning, producing a pruned model f_P^{i+n} . Since pruning typically degrades accuracy, the pruned network subsequently restores performance using the same KD procedure employed in *Stage 1*. The distilled-and-recovered student can either be used as the final DAC model or re-enter the loop for additional pruning cycles, enabling up to N iterations. Details in Chapter III-C.

¹**GitHub Repository:** <https://github.com/RicoStaedeli/DAC-Loop>

II. RELATED WORK

To enable deployment under tight compute and memory budgets, prior work in RS has explored lightweight architectures and compression of standard CNN backbones [3, 2]. Model compression via pruning and quantization typically exposes an accuracy–efficiency trade-off. Stronger reductions in parameters often lead to measurable performance drops [8, 9]. For instance, Shinde et al. [8] prune and quantize CNNs on EuroSAT and report substantial efficiency gains with limited accuracy degradation.

Teacher–student KD has been widely adopted to improve compact RS classifiers, frequently enabling small students to approach the accuracy of larger teachers [4–6]. However, KD is commonly used as a training technique for fixed student architectures, rather than as a mechanism to support *progressive* model compression. Outside RS, several prior studies use KD to recover accuracy after pruning. Aghli et al. [10] combine pruning and KD by using a pruned model in the teacher–student setup, and Muralidharan et al. [11] show that iterative structured pruning with KD-based retraining can yield compact yet accurate models. In RS scene classification, KD and pruning are still most often evaluated separately or in one-shot combinations [4, 8, 9]. In contrast, DAC-Loop studies *iterative* RS compression with (i) *KD* applied before and after structured channel pruning and (ii) a scheduled alternation of structured and unstructured pruning within a repeatable protocol.

III. METHODOLOGY

A. Knowledge Distillation

In the first KD stage the predictive knowledge of a teacher model is transferred to a smaller student. Following Hinton et al. [12], the student is trained using a hybrid distillation objective. Let T_ϕ and S_θ denote the teacher and student networks, respectively. The KD objective is defined as

$$L = (1 - \alpha) L_{\text{CE}}(y, S_\theta(x)) + \alpha L_{\text{KD}}(T_\phi(x), S_\theta(x)), \quad (1)$$

where α controls the trade-off between supervised learning and teacher guidance. To select the model individual best suiting KD method, four different KD-strategies were implemented. These methods differ in the definition of L_{KD} . A special case is the *KD via Cosine Embedding Loss* variant. For this method the *teacher* and *student* feature representations must have matching dimensionality, since the cosine loss is defined on pairs of vectors in the same space. Let $\mathbf{f}^{(t)} \in \mathbb{R}^{d_t}$ and $\mathbf{f}^{(s)} \in \mathbb{R}^{d_s}$ denote the intermediate feature vectors of the teacher and student, respectively. To enable a well-defined similarity computation when $d_t \neq d_s$, a learnable projection layer was introduced which maps the student features into the teacher’s feature space. The resulting *cosine-embedding KD loss* is then computed between $\mathbf{f}^{(t)}$ and $\mathbf{f}_{\text{proj}}^{(s)}$, ensuring shape compatibility while still allowing architectural flexibility between teacher and student models. A hyperparameter search is conducted to identify the best-performing individual strategies for each model, which are then used in subsequent stages of the DAC loop.

B. Pruning

To reduce the parameter count, we apply two complementary strategies: (1) global unstructured (weight-level) pruning and (2) structured channel pruning for convolutional layers.

1) *Global Unstructured Pruning*: Let $\{\mathbf{W}_\ell\}_{\ell=1}^L$ be the weight tensors of all prunable layers (convolutional and linear). We associate binary masks $\{\mathbf{M}_\ell\}_{\ell=1}^L$ of identical shape and define masked weights

$$\widetilde{\mathbf{W}}_\ell = \mathbf{M}_\ell \odot \mathbf{W}_\ell, \quad (2)$$

where $\odot$ denotes elementwise multiplication and $(\mathbf{M}_\ell)_i \in \{0, 1\}$ indicates whether parameter i is kept (1) or pruned (0). For a target global sparsity $\alpha \in [0, 1]$ (fraction of pruned weights across all prunable layers), we form the multiset $\mathcal{W}$ of all scalar weights. In magnitude pruning, we choose a global threshold τ such that a fraction α of $w \in \mathcal{W}$ satisfies $|w| \leq \tau$, and set

$$(\mathbf{M}_\ell)_i = \begin{cases} 0, & \text{if } |(\mathbf{W}_\ell)_i| \leq \tau, \\ 1, & \text{otherwise.} \end{cases} \quad (3)$$

Masks are applied via re-parameterization during training so pruned weights do not contribute to the forward pass.

2) *Structured Channel Pruning*: While global unstructured pruning increases weight-level sparsity, it does not change the layer shapes and thus does not directly reduce the number of channels. To obtain a structurally smaller network, we additionally perform structured pruning of convolutional channels following the channel-pruning formulations introduced by Li et al. [13] and He et al. [14]. Consider a convolutional layer ℓ with weight tensor

$$\mathbf{W}_\ell \in \mathbb{R}^{C_{\text{out}} \times C_{\text{in}} \times k_h \times k_w}, \quad (4)$$

where each slice $\mathbf{W}_{\ell,i,:,:,}$ corresponds to the kernel associated with output channel $i \in \{1, \dots, C_{\text{out}}\}$. We define the importance score of channel i as the ℓ_2 norm of its kernel:

$$s_{\ell,i} = \|\mathbf{W}_{\ell,i,:,:,}\|_2^2 = \sum_{c=1}^{C_{\text{in}}} \sum_{u=1}^{k_h} \sum_{v=1}^{k_w} (\mathbf{W}_{\ell,i,c,u,v})^2. \quad (5)$$

For a defined pruning ratio $r_\ell \in [0, 1]$ for layer ℓ , we compute $n_\ell = \lfloor r_\ell C_{\text{out}} \rfloor$ and prune the n_ℓ output channels with the smallest scores $s_{\ell,i}$. In practice, removing channels from one layer affects the input dimensionality of subsequent layers. To ensure architectural consistency, a dependency graph of the network is constructed. Pruning is then applied to the entire group atomically, guaranteeing that all tensor shapes remain compatible and the resulting network is a valid, smaller architecture.

C. Distillation–Aware Compression Loop (DAC-Loop)

As illustrated in Fig. 1, a *teacher* model $f_T(\cdot; \theta_T)$ is employed, instantiated as a ResNet50. The model f_T is pretrained once and kept frozen throughout the entire DAC-Loop, therefore the same teacher is reused at every iteration. A smaller *student* network $f_S^{i+n}(\cdot; \theta_S)$, in this work a ResNet18, is randomly initialized and trained with the defined KD-method and the pretrained *teacher* model f_T in *Stage 1*. The

TABLE I
KD-1 & KD-2 BOTH REPRESENT **STAGE 1 TRAINING WITH KD** IN OUR DAC-LOOP (FIGURE 1) AND INDICATE WHETHER KD IS APPLIED PRE (KD-1) OR POST (KD-2) PRUNING. **N** DENOTES THE NUMBER OF DAC-LOOP ITERATIONS.

Ablation Study Results for DAC-Loop with a ResNet18									
	Model	KD-1	KD-2	N	Compression	Parameters	Prun. Type	Val. Acc. (EuroSAT)	Val. Acc. (RESISC45)
KD	ResNetTiny	✓	–	1	0%	1.0 M	–	96.49± 0.17	82.63± 0.32
	ResNet10mini	✓	–	1	0%	2.0 M	–	97.11± 0.15	86.36± 0.37
	ResNet18	✓	–	1	0%	11.2 M	–	98.61± 0.08	91.95± 0.17
Unstr.	ResNet18	✓	–	1	0%	11.2 M	Unstructured	98.58± 0.06	92.59± 0.02
	ResNet18	–	✓	1	0%	11.2 M	Unstructured	98.41± 0.02	92.27± 0.20
	ResNet18	✓	✓	1	0%	11.2 M	Unstructured	98.67± 0.05	92.22± 0.03
Comp. +	ResNet18	–	–	1	46.0%	6.0 M	Structured	98.24± 0.02	90.78± 0.11
	ResNet18	✓	–	1	46.0%	6.0 M	Structured	98.37± 0.06	82.02± 0.15
	ResNet18	–	✓	1	46.0%	6.0 M	Structured	98.46± 0.04	92.10± 0.04
	ResNet18	✓	✓	1	46.0%	6.0 M	Structured	98.48± 0.06	92.22± 0.03
Comp. ++	ResNet18	–	–	1	75.0%	2.8 M	Structured	97.82± 0.10	87.86± 0.17
	ResNet18	✓	–	1	75.0%	2.8 M	Structured	98.20± 0.04	89.98± 0.29
	ResNet18	–	✓	1	75.0%	2.8 M	Structured	98.09± 0.06	90.13± 0.20
	ResNet18	✓	✓	1	75.0%	2.8 M	Structured	98.41± 0.16	91.11± 0.08
	ResNet18	✓	✓	7	84.7%	1.7 M	both	97.63± 0.11	91.61± 0.24
Loop	ResNet18	✓	✓	9	91.7%	0.9 M	both	96.97± 0.46	90.64± 0.15

distilled student model f_S^{i+n} obtained from Stage 1 is then passed to *Stage 2*, where it is compressed using structured or unstructured pruning yielding the model f_P^{i+1} . This requires a subsequent fine-tuning step to recover performance, which is again performed using the KD-based training procedure from Stage 1, where the compressed model continues to benefit from the same teacher’s guidance. After pruning and KD recovery, the student f_S^{i+n} is either used as output DAC-Model or enters again Stage 2, where it is pruned again. This loop allows the process to run for N compression cycles. To support multiple DAC iterations, we employ a compression scheduler that assigns specific pruning strategies, hyperparameter to each loop cycle. This enables different compression across iterations, e.g. unstructured pruning in the first cycle and structured pruning in the next.

IV. EXPERIMENTS & RESULTS

A. Data & Setup

To evaluate the proposed DAC-Loop the RS datasets *EuroSAT* [15] and *NWPU-RESISC45* [16] were used. To train the models, we adopt a 60-20-20 split. Normalization is applied across all experiments, while data augmentation is employed only for training the ResNet50 teacher model. All experiments were conducted on a Tesla V100-SXM3-32GB. We trained all models using the AdamW optimizer with an initial learning rate of 3×10^{-4} and a batch size of 256. Unless stated otherwise, the learning rate was scheduled using a cosine annealing schedule over the full 50 training epochs.

B. Baselines

Table II reports the model baselines we trained from scratch and compares them to published baselines in prior work [17, 18, 15]. The results show that our implementations reach comparable accuracy on both EuroSAT and NWPU-RESISC45, providing a reliable reference point for the compression experiments that follow. All our experiments are conducted with four different seeds to assess the robustness

of the results. Except for the ResNet50 teacher model, all baselines were trained without data augmentation, while the teacher was trained with data augmentation for 100 epochs.

TABLE II
 BASELINE VALIDATION ACCURACY

Model	Param	EuroSAT	RESISC45
<i>Baselines - others</i>			
ResNet18 [17]	11.7 M	98.06	92.10
ResNet50 [18]	24.0 M	99.20	96.80
ResNet50 [15]	24.0 M	98.57	–
<i>Baselines - ours</i>			
ResNetTiny	0.9 M	95.77± 0.06	80.15± 0.54
ResNet10mini	2.0 M	96.41± 0.09	83.43± 0.44
ResNet10	4.9 M	98.37± 0.05	88.33± 0.19
ResNet18	11.2 M	98.34± 0.05	89.76± 0.14
ResNet50 - Teacher	24.6 M	96.79± 0.18	92.44± 0.15

C. Ablation Study

In Table I, the results of our DAC-Loop ablation study are grouped into five blocks: *KD* (no compression, only KD), *Comp. +* (moderate structured compression strength at 46.0% parameter reduction), *Unstr.* (unstructured compression), *Comp. ++* (stronger structured compression strength at 75.0%), and *Loop* (multiple DAC-Loop iterations). Within each block, we systematically test the KD components or vary the number of iterations N . Across both one-shot structured compression degrees (*Comp. +* and *Comp. ++*), the grey-highlighted rows show that using KD in both stages of our DAC-Loop yields the best performance. This confirms that integrating pre-pruning KD with post-pruning KD in structured channel pruning is more effective than either approach alone. The *Loop* experiments further illustrate the benefit of iterating the DAC-Loop: with $N = 7$ iterations we obtain a model compressed to 1.7M parameters that still achieves an accuracy of 91.61% on the NWPU-RESISC45 Dataset (bold in Table I). This ablation study demonstrates

that repeated DAC-Loop cycles enable progressively higher compression while preserving, and even enhancing, model accuracy. It can also be observed that the results on both the NWPU-RESISC45 and EuroSAT datasets consistently confirm this finding. However, the improvement obtained with DAC-Loop is substantially more pronounced on NWPU-RESISC45.

Table I shows that applying global unstructured magnitude pruning boosts accuracy to 92.22%, outperforming only using KD. Because unstructured pruning removes individual weights rather than whole channels, it leaves the architecture and parameter count unchanged. While this sparsity doesn't compress the model, it helped boost accuracy beyond what structured pruning alone achieves. Our empirical results show that the DAC-Loop achieves the best performance when both complementary pruning methods are integrated into a unified framework. In the ablation study, the compression results obtained with DAC-Loop (last two rows of Table I) are produced using the pruning schedule summarized in Table III.

TABLE III
PRUNING SCHEDULE PARAMETERS AND VALIDATION ACCURACY FOR RESNET18 ON NWPU-RESISC45

Iteration (N)	Pruning Type	Parameter	Val. Acc.
1	Unstructured	11.2 M	92.18 $\pm$ 0.06
2	Structured	6.0 M	92.41 $\pm$ 0.11
3	Unstructured	6.0 M	92.53 $\pm$ 0.12
4	Structured	3.2 M	92.18 $\pm$ 0.09
5	Unstructured	3.2 M	92.21 $\pm$ 0.08
6	Structured	1.7 M	91.54 $\pm$ 0.14
7	Unstructured	1.7 M	91.61 $\pm$ 0.24
8	Structured	0.9 M	90.58 $\pm$ 0.11
9	Unstructured	0.9 M	90.64 $\pm$ 0.16

D. Results and Discussion

Figure 2 illustrates the effect of the DAC-Loop on the accuracy-compression trade-off for ResNet18 on NWPU-RESISC45. In this experiment we employ the same ResNet18 with an without the DAC-Loop. Across the entire compression range, the DAC-Loop curve remains consistently above the purely one-shot pruned baseline, demonstrating that distillation-guided recovery stabilizes accuracy. The baseline (blue in Fig. 2) consist of one-shot compression with varying pruning levels, followed by a subsequent stage of supervised fine-tuning. The difference persists across all compression rates between 10% and 90%. This confirms that the DAC-Loop does not only mitigate pruning loss, it fundamentally shifts the trade-off, enabling significantly smaller models to maintain strong performance.

Additionally, Table IV shows that at 256 $\times$ 256 input resolution, the DAC-Loop reduces dense compute from 9.02G to 0.81G MACs (11.16 $\times$) and achieves a 4.42 $\times$ inference speedup on a Tesla V100-SXM3-32GB with a batch size of 256.

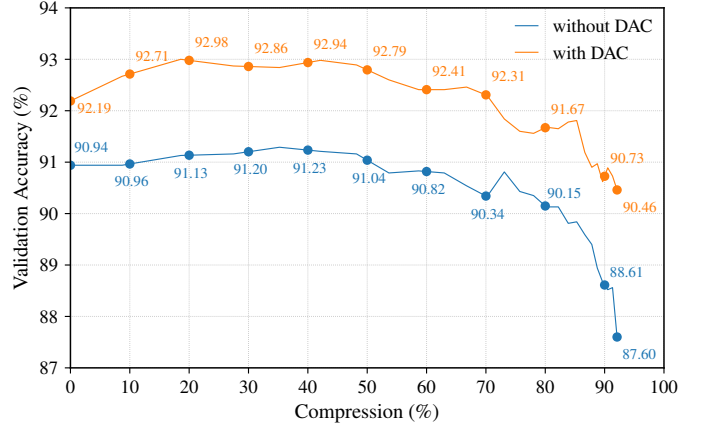


Fig. 2. **Effect of DAC-Loop on the accuracy-compression trade-off:** comparison of validation accuracy on NWPU-RESISC45 under increasing compression for ResNet18 with (orange) and without (blue) the proposed DAC-Loop. Across all compression levels, the DAC-Loop maintains consistently higher accuracy than one-shot pruning.

TABLE IV
EFFICIENCY COMPARISON OF THE FULL AND DAC-LOOP-COMPRESSED RESNET18 MODELS ON THE RESISC45 VALIDATION SET.

Metric	Full	Compressed	Speedup
Parameters $\downarrow$	11.2 M	0.9 M	$\times 12.44$
FLOPs $\downarrow$	18.05 G	1.62 G	$\times 11.16$
MACs $\downarrow$	9.02 G	0.81 G	$\times 11.16$
Inference (Batch) $\downarrow$	288.25 ms	65.18 ms	$\times 4.42$

V. CONCLUSION

We introduced the Distillation-Aware Compression Loop (DAC-Loop), which is a protocol that combines knowledge distillation with structured and unstructured pruning for remote-sensing CNNs. The loop alternates KD-based training and pruning, guided by a compression scheduler that selects pruning type and sparsity per iteration, progressively compressing students while realigning them with the teacher.

On EuroSAT and NWPU-RESISC45, the DAC-Loop yields compact, high-performing ResNet students. For a ResNet18 student on NWPU-RESISC45, it reduces parameters by over 90% while matching or exceeding the supervised baseline. Ablations show that applying KD before and after structured channel pruning consistently outperforms pure pruning, and that iterative DAC cycles produce highly compressed models that retain strong accuracy. Global unstructured magnitude pruning further improves generalization without altering the architecture, and combining it with structured pruning inside the DAC-Loop improves regularization and true architectural compression. The compression-accuracy curves in Figure 2 confirm that the DAC-Loop shifts the trade-off toward smaller models, with DAC-compressed networks remaining more accurate than purely pruned counterparts across the full compression range. Overall, our study of iterative pruning in combination with KD shows promising results to generate small yet high performing models for RS applications.

REFERENCES

- [1] Z. Xiong, F. Zhang, Y. Wang, Y. Shi, and X. X. Zhu, "EarthNets: Empowering AI in Earth Observation," Apr. 2024, arXiv:2210.04936 [cs]. [Online]. Available: <http://arxiv.org/abs/2210.04936>
- [2] Y. Zhao, S. Li, and X. Feng, "Lightweight Remote Sensing Scene Classification on Edge Devices via Knowledge Distillation and Early-exit," Jul. 2025, arXiv:2507.20623 [cs]. [Online]. Available: <http://arxiv.org/abs/2507.20623>
- [3] B. Liu, C. Zhan, C. Guo, X. Liu, and S. Ruan, "Efficient remote sensing image classification using the novel STConvNeXt convolutional network," *Scientific Reports*, vol. 15, no. 1, p. 8406, Mar. 2025. [Online]. Available: <https://www.nature.com/articles/s41598-025-92629-x>
- [4] H. Song, J. Xie, L. Liang, Y. Su, Y. Xiao, X. Zhang, Y. Ouyang, X. Li, S. Chen, and Y. Li, "Symmetrical Learning and Transferring: Efficient Knowledge Distillation for Remote Sensing Image Classification," *Symmetry*, vol. 17, no. 7, p. 1002, Jun. 2025. [Online]. Available: <https://www.mdpi.com/2073-8994/17/7/1002>
- [5] Y. He, J. Cai, Q. Hu, and P. Wang, "Remote Sensing Image Classification with Decoupled Knowledge Distillation," 2025, version Number: 2. [Online]. Available: <https://arxiv.org/abs/2505.19111>
- [6] Y. Himeur, N. Aburaed, O. Elharrouss, I. Varlamis, S. Atalla, W. Mansoor, and H. A. Ahmad, "Applications of Knowledge Distillation in Remote Sensing: A Survey," 2024, version Number: 1. [Online]. Available: <https://arxiv.org/abs/2409.12111>
- [7] M. M. H. Shuvo, S. K. Islam, J. Cheng, and B. I. Morshed, "Efficient Acceleration of Deep Learning Inference on Resource-Constrained Edge Devices: A Review," *Proceedings of the IEEE*, vol. 111, no. 1, pp. 42–91, Jan. 2023. [Online]. Available: <https://ieeexplore.ieee.org/document/9985008/>
- [8] T. Shinde, "Model Compression Meets Resolution Scaling for Efficient Remote Sensing Classification," in *2025 IEEE/CVF Winter Conference on Applications of Computer Vision Workshops (WACVW)*. Tucson, AZ, USA: IEEE, Feb. 2025, pp. 1576–1584. [Online]. Available: <https://ieeexplore.ieee.org/document/10972523/>
- [9] G. Zou, L. Yao, F. Liu, C. Zhang, X. Li, N. Chen, S. Xu, and J. Zhou, "RemoteTrimmer: Adaptive Structural Pruning for Remote Sensing Image Classification," Dec. 2024, arXiv:2412.12603 [cs]. [Online]. Available: <http://arxiv.org/abs/2412.12603>
- [10] N. Aghli and E. Ribeiro, "Combining Weight Pruning and Knowledge Distillation For CNN Compression," in *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*. Nashville, TN, USA: IEEE, Jun. 2021, pp. 3185–3192.
- [11] S. Muralidharan, S. T. Sreenivas, R. Joshi, M. Chochowski, M. Patwary, M. Shoybi, B. Catanzaro, J. Kautz, and P. Molchanov, "Compact Language Models via Pruning and Knowledge Distillation," Nov. 2024, arXiv:2407.14679 [cs]. [Online]. Available: <http://arxiv.org/abs/2407.14679>
- [12] G. Hinton, O. Vinyals, and J. Dean, "Distilling the Knowledge in a Neural Network," Mar. 2015, arXiv:1503.02531 [stat]. [Online]. Available: <http://arxiv.org/abs/1503.02531>
- [13] H. Li, A. Kadav, I. Durdanovic, H. Samet, and H. P. Graf, "Pruning Filters for Efficient ConvNets," Mar. 2017, arXiv:1608.08710 [cs]. [Online]. Available: <http://arxiv.org/abs/1608.08710>
- [14] Y. He, X. Zhang, and J. Sun, "Channel Pruning for Accelerating Very Deep Neural Networks," Aug. 2017, arXiv:1707.06168 [cs]. [Online]. Available: <http://arxiv.org/abs/1707.06168>
- [15] P. Helber, B. Bischke, A. Dengel, and D. Borth, "EuroSAT: A Novel Dataset and Deep Learning Benchmark for Land Use and Land Cover Classification," Feb. 2019, arXiv:1709.00029 [cs]. [Online]. Available: <http://arxiv.org/abs/1709.00029>
- [16] G. Cheng, J. Han, and X. Lu, "Remote Sensing Image Scene Classification: Benchmark and State of the Art," *Proceedings of the IEEE*, vol. 105, no. 10, pp. 1865–1883, Oct. 2017. [Online]. Available: <http://ieeexplore.ieee.org/document/7891544/>
- [17] A. Sharma, "Improving Remote Sensing Classification using Topological Data Analysis and Convolutional Neural Networks," Jul. 2025, arXiv:2507.10381 [cs]. [Online]. Available: <http://arxiv.org/abs/2507.10381>
- [18] M. Neumann, A. S. Pinto, X. Zhai, and N. Houlsby, "In-domain representation learning for remote sensing," Nov. 2019, arXiv:1911.06721 [cs]. [Online]. Available: <http://arxiv.org/abs/1911.06721>