Diversity-Based Two-Phase Pruning Strategy for Maximizing Image Segmentation Generalization with Applications in Transmission Electron Microscopy

Ze-wei Ye, Hung-Wei Hsueh, Shu-han Hsu*
Department of Computer Science and Information Engineering, National Cheng Kung University
*shhsu@gs.ncku.edu.tw

Abstract

To address the storage and computational demands of Transmission Electron Microscopy (TEM), we propose a two-phase pruning strategy that reduces model size and enhances speed while maintaining performance across diverse datasets, practical for TEM analysis. Unlike traditional pruning methods that focus solely on weight magnitude, our approach also considers weight variability to preserve feature diversity, crucial for generalization in the varied context of TEM images. Our strategy first prunes filters with low magnitude and variability, then removes redundant filters with high linear similarity. This two-phase pruning, followed by fine-tuning, effectively reduces parameters and computational load while ensuring high accuracy and generalizability.

1 Introduction

Transmission electron microscopy (TEM) is essential for observing materials at the nanoscale, but traditional manual segmentation of TEM images is time-consuming and error-prone. Recent deep learning advancements have enhanced automated segmentation, enabling efficient identification of nanoscale features [1, 2, 3]. However, deep learning for TEM analysis faces challenges due to limited TEM hardware memory capacity, necessitating pruning techniques to reduce computational demands while maintaining accuracy. Preserving filter diversity is crucial for effective image segmentation [4, 5], as high filter similarity can impair model generalization. Thus, maintaining diversity in filter weights while minimizing parameter count is key. Please refer to Appendix A for further details.

To address these challenges, we propose a Diversity-Based Two-Phase Pruning Strategy that balances weight diversity preservation with parameter reduction, optimizing model performance for TEM image analysis.

2 Related Work

Ref. [6] introduced magnitude-based filter pruning, using the L1-norm of all kernels to assess filter importance and removing those below a threshold, including their corresponding kernels in subsequent layers. Their strategy for residual blocks considers only the weights of projection shortcut connections (1x1 kernels). Our approach expands this by including the weights of subsequent layers with 3x3 kernels, enhancing diversity in feature extraction.

Ref. [7] evaluates filters and kernels as high-dimensional vectors rather than scalars. We apply this perspective to assess filter capability based on the diversity among kernel vectors, recognizing that greater diversity improves generalization in image segmentation and recognition tasks.

38th Conference on Neural Information Processing Systems (NeurIPS 2024).

Ref. [8, 9] identify high similarity in convolutional filters as a source of redundancy, showing that ignoring similarity leads to accuracy loss. Our method addresses this by retaining only the filter with the largest L1-norm among similar filters, eliminating redundant weights. Unlike pairwise pruning [9], our approach groups all highly correlated filters, retaining only the one with the highest L1-norm.

The sequence of pruning is also critical. Ref. [10] demonstrates that pruning by weight importance first, followed by similarity, minimizes accuracy loss. Unlike the hybrid pruning in ref. [10], which emphasizes scalar weights and their differences, our approach integrates the geometric implications of filters and kernels in high-dimensional vector space, enabling a more comprehensive pruning strategy that enhances generalization capabilities and complexity-performance trade-off.

3 Methods

3.1 Backbone Model Structure

For TEM nanoparticle detection, we use a residual U-Net with three encoder-decoder layers and a bottleneck, each with two residual blocks for faster convergence and improved feature extraction. Structured pruning [11, 12] removes redundant filters, adjusting connected kernels in subsequent layers [6], resulting in significant model compression and reduced inference time.

For layers with residual shortcuts, filters are pruned at matching indices to maintain output alignment. We evaluated filter magnitude by two methods: using shortcut weights alone [6] and by summing weights from both the residual block and shortcut layers. Summing yielded higher accuracy, which we use for all pruning evaluations in shortcut-connected layers.

3.2 Phase 1: Quantitative Score Pruning

Studies have demonstrated that filter magnitude significantly affects model performance, necessitating its use as an evaluation metric. Accordingly, the L1-norms of all kernels within a filter are summed to quantify its contribution to feature extraction [6].

For kernels larger than 1×1, multiplication with input feature maps can be seen as a linear transformation in a high-dimensional space. For example, 3×3 kernels are reshaped into 1×9 vectors [7], where greater diversity among these vectors indicates better filter generalization for feature extraction.

Two metrics are proposed to evaluate diversity within this vector space: (1) variance in the magnitudes (L2-norms) of kernel vectors, reflecting differences in feature intensity, and (2) variance in their directional diversity, measured by the Euclidean distance between each kernel vector and the mean kernel vector within a filter.

To standardize these metrics, min-max normalization scales them to the [0,1] range. The overall significance score for each filter is calculated as:

$$I_p = \|F_p\|_1 + var_q\left(\|K_{p,q}\|_2 \mid q = 1, 2, \ldots, Q\right) + var_q\left(\|K_{p,q} - K_{p,avg}\|_2 \mid q = 1, 2, \ldots, Q\right)$$

where F_p denotes the p-th filter within a given layer, $K_{p,q}$ represents the q-th kernel within the p-th filter. The average kernel of the p-th filter is denoted as $K_{p,avg}$. Q denotes the total number of kernels contained in the filter.

For filters with a single kernel or 1×1 kernels, the significance score is based solely on magnitude ($\|F_p\|_1$). Filters are ranked by significance scores, and a proportion of 1-pruning ratio is retained according to the pruning strategy.

3.3 Phase 2: Similarity Pruning

After Phase 1 pruning, the remaining filters generally have higher magnitudes and greater diversity, but some may still exhibit similar linear relationships, resulting in redundant outputs [8, 9, 13, 14, 15]. To further reduce computational burden, we remove these similar filters.

First, each filter is represented as a single vector by averaging its kernels. The Pearson correlation coefficient is then used to measure similarity between these vectors [9, 13, 16], with values closer to ±1 indicating stronger correlation. The coefficient is calculated as:

$$r_{X,Y} = cov(X,Y)/\sigma_X\sigma_Y$$

where $cov(X,Y)$ is covariance between vectors X and Y, and σ_X is standard deviation for vector X.

Filters with a correlation coefficient above a set threshold are considered redundant. The filter with the smaller $\|F_p\|_1$ is removed, retaining the one with the larger L1-norm. This ensures greater diversity among the remaining filters. The model is then fine-tuned to restore accuracy.

Illustrations, further details of the methods, and ablation study are provided in Appendix B.1 and D.

4 Experiments

4.1 Setup and Dataset

All tests used TensorFlow 2.14 on an Nvidia RTX A5000 (24GB) GPU and an AMD EPYC 7313 16-Core CPU. The TEM dataset [1] includes images of 2, 5, 10, and 20 nm gold nanoparticles, with 4,608 training and 1,352 validation images, randomly shuffled across particle sizes.

We applied our Diversity-Based Two-Phase Pruning Strategy to a pre-trained residual U-Net (4,012,963 parameters, 15.31 MB, 96.33% initial accuracy). In both phases, one-shot pruning across all layers was followed by fine-tuning with a gradually reduced learning rate to limit accuracy loss to under 1%. Details are in Appendix B.2.

Computational complexity was evaluated using Multiply-Accumulate Operations (MACs), a standard metric representing multiplication and addition operations required in convolutional and fully connected layers.

4.2 Results

Table 1: Performance of the diversity-based two-phase pruned model compared to the original model

	Accuracy	**Parameters**	**MACs**	**Prediction (GPU)**	**Prediction (CPU)**
Original	96.33%	4012963	$\approx 79G$	16.27ms / image	347.63ms / image
Pruned	95.50%	210097	$\approx 5.764G$	5.92ms / image	99.11ms / image
Reduction	**0.83%**	**95%**	**92.70%**	**63.61%**	**71.49%**

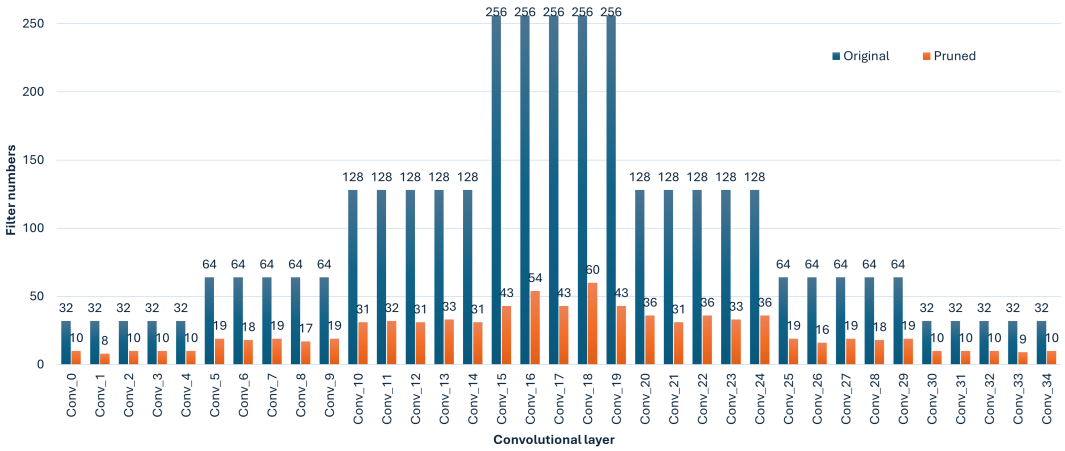

Figure 1: Filter count per layer in residual U-Net before and after diversity-based two-phase pruning.

With 70% pruning in Phase 1 and a 0.8 Pearson correlation threshold in Phase 2, the model is reduced to 210,097 parameters (820.69 KB), approximately 5% of its original size, as shown in Table 1. It achieves 95.50% validation accuracy, with a 92.7% MAC reduction and a 0.83% accuracy drop,

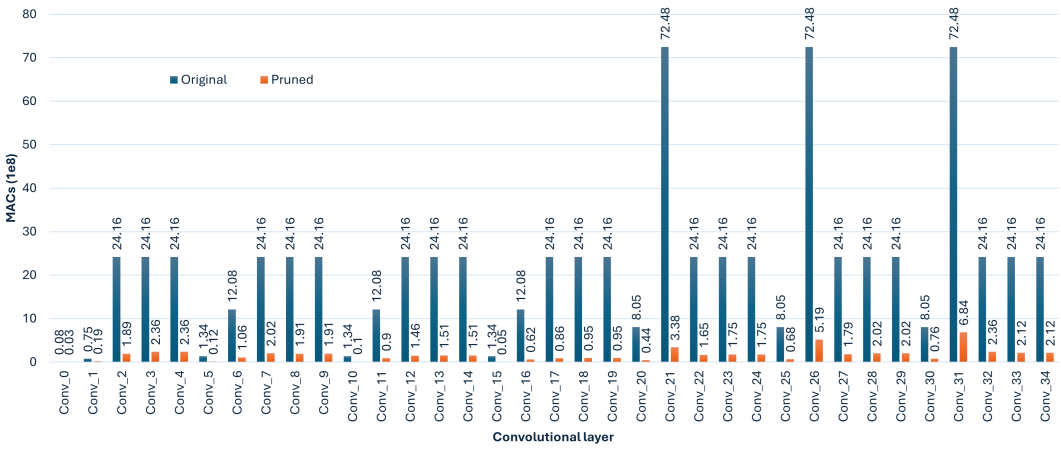

Figure 2: MACs per layer of the residual U-Net before and after diversity-based two-phase pruning.

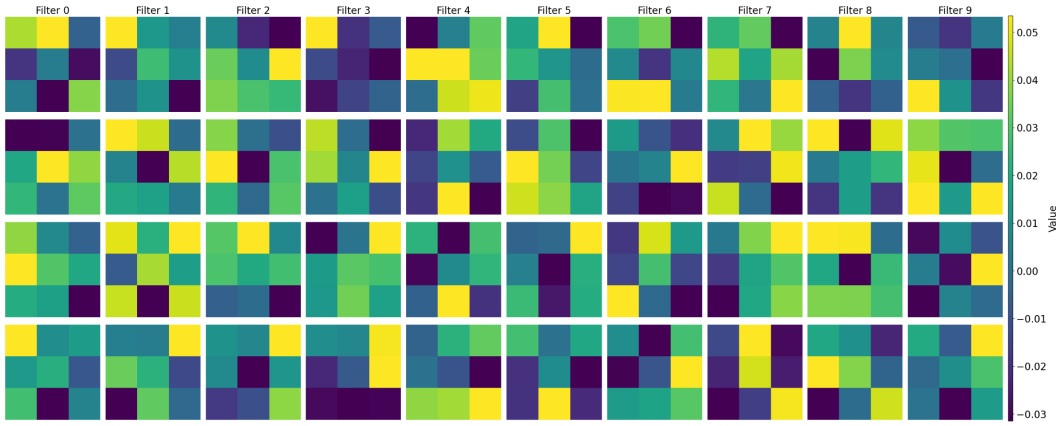

Figure 3: Heat maps of filters in four layers showing notable differences among the retained filters.

within the 1% threshold. Fig. 1 and 2 illustrate filters and MACs before and after pruning. Detailed comparisons are in Appendix B.2, with pruned performance across various ratios summarized in Appendix C.

This method significantly reduces parameters and computational load with minimal accuracy loss. Fig. 3 shows heat maps of pruned filters in four layers, each retaining 10 diverse filters. The structured pruning maintains the full 4D weight matrix, enhancing GPU efficiency and reducing inference time. Appendix E compares this approach with state-of-the-art pruning methods, demonstrating its competitiveness.

5 Conclusion and Future Work

We propose a two-phase pruning strategy to overcome limitations of magnitude-based pruning in image recognition and segmentation, preserving filter diversity while reducing parameters. Phase 1 retains filters with high magnitude and diversity; Phase 2 prunes redundant filters based on Pearson correlation. After pruning and fine-tuning, the residual U-Net maintained strong performance on TEM images of gold nanoparticles. This approach enhances model generalization with minimal computational cost and without retraining on new samples, provided the initial training dataset is well-chosen. Ongoing work includes adopting different fine-tuning strategies, optimizing thresholds for each layer, and even pruning at a finer granularity to achieve higher pruning precision and better performance. This study shows that diversity-focused pruning effectively reduces parameters while preserving accuracy, thereby accelerating TEM image analysis for materials characterization and reducing computational storage requirements.

Acknowledgements

Thank you to the reviewers for their detailed feedback allowing us to improve and clarify our work. We are also grateful for the support from the National Science and Technology Council (Taiwan, ROC) through Project NSTC 112-2221-E-006 -151 -MY3 and Google Research Scholar Award.

References

[1] K. Sytwu, C. Groschner, and M. C. Scott, “Understanding the Influence of Receptive Field and Network Complexity in Neural Network-Guided TEM Image Analysis,” *Microscopy and Microanalysis*, vol. 28, no. 6, pp. 1896–1904, 2022.

[2] J. P. Horwath, D. N. Zakharov, R. Mégret, et al., "Understanding important features of deep learning models for segmentation of high-resolution transmission electron microscopy images," *npj Computational Materials.*, vol. 6, no. 1, p. 108, 2020.

[3] K. Faraz, T. Grenier, C. Ducottet, et al., "Deep learning detection of nanoparticles and multiple object tracking of their dynamic evolution during in situ ETEM studies," *Scientific Reports*, vol. 12, p. 2484, 2022.

[4] G. E. Hinton, N. Srivastava, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, "Improving neural networks by preventing co-adaptation of feature detectors," arXiv preprint arXiv:1207.0580, 2012.

[5] M. Hafizur and M. H. R. Masum, "Visualizing and understanding convolutional networks," in *Computer Vision—ECCV 2014*, pp. 818–833, 2014.

[6] H. Li, A. Kadav, I. Durdanovic, H. Samet, and H. P. Graf, "Pruning Filters for Efficient ConvNets," in *International Conference on Learning Representations (ICLR)*, 2017.

[7] V K. Koo and H. Kim, "V-SKP: Vectorized Kernel-Based Structured Kernel Pruning for Accelerating Deep Convolutional Neural Networks," *IEEE Access*, vol. 11, pp. 118547-118557, 2023.

[8] B. Ayinde, T. Inanc, and J. Zurada, "Redundant feature pruning for accelerated inference in deep neural networks," *Neural Networks*, vol. 118, pp. 102–111, Apr. 2019.

[9] P. Singh, V. Verma, P. Rai and V. Namboodiri, "Leveraging Filter Correlations for Deep Model Compression," in *2020 IEEE Winter Conference on Applications of Computer Vision (WACV)*, Snowmass Village, CO, USA, 2020, pp. 824-833.

[10] P. Thaker and B. Mohan, "Comparing Different Sequences of Pruning Algorithms for Hybrid Pruning," in *Proceedings of the 14th International Conference on Computing, Communication and Networking Technologies (ICCCNT)*, 2023, pp. 1-7.

[11] H. Cheng, M. Zhang and J. Q. Shi, "A Survey on Deep Neural Network Pruning: Taxonomy, Comparison, Analysis, and Recommendations," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2024.

[12] T. Liang, J. Glossner, L. Wang, S. Shi, and X. Zhang, "Pruning and quantization for deep neural network acceleration: A survey," *Neurocomputing*, vol. 461, pp. 370–403, Oct. 2021.

[13] S. H. Shabbeer Basha, S. N. Gowda and J. Dakala, "A Simple Hybrid Filter Pruning for Efficient Edge Inference," in *ICASSP 2022 - 2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, Singapore, 2022, pp. 3398-3402.

[14] L. Geng and B. Niu, "Pruning convolutional neural networks via filter similarity analysis," *Machine Learning*, vol. 111, pp. 1-20, 2022.

[15] W. Wang, Z. Yu, C. Fu, D. Cai, and X. He, "COP: Customized Correlation-Based Filter Level Pruning Method for Deep CNN Compression," *Neurocomputing*, vol. 464, pp. 533–545, Nov. 2021.

[16] P. Molchanov, A. Mallya, S. Tyree, I. Frosio and J. Kautz, "Importance Estimation for Neural Network Pruning," in *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, Long Beach, CA, USA, 2019, pp. 11256-11264.

[17] F. N. Iandola, S. Han, M. W. Moskewicz, K. Ashraf, W. J. Dally, and K. Keutzer, "SqueezeNet: AlexNet-level accuracy with 50x fewer parameters and <0.5MB model size," arXiv:1602.07360, 2016.

[18] N. Beheshti and L. Johnsson, "Squeeze U-Net: A Memory and Energy Efficient Image Segmentation Network," *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, Seattle, WA, USA, 2020, pp. 1495-1504.

[19] C. Dayananda, J. Y. Choi and B. Lee, "A Squeeze U-SegNet Architecture Based on Residual Convolution for Brain MRI Segmentation," in *IEEE Access*, vol. 10, pp. 52804-52817, 2022.

[20] M. Lin, R. Ji, Y. Zhang, B. Zhang, Y. Wu, and Y. Tian, "Channel pruning via automatic structure search," 2020, arXiv:2001.08565.

[21] S. Zhong, G. Zhang, N. Huang, and S. Xu, "Revisit kernel pruning with lottery regulated grouped convolutions," in *Proc. Int. Conf. Learn. Represent.*, 2022, pp. 1–23.

[22] M.Lin,R.Ji,Y.Wang,Y.Zhang,B.Zhang,Y.Tian,andL.Shao,"HRank: Filter pruning using high-rank feature map," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2020, pp. 1529–1538.

[23] Z.Wang,S.Lin,J. Xie, and Y. Lin, "Pruning blocks for CNN compression and acceleration via online ensemble distillation," *IEEE Access*, vol. 7, pp. 175703–175716, 2019.

[24] S.Lin,R.Ji, C. Yan, B.Zhang,L.Cao,Q.Ye,F.Huang,andD. Doermann, "Towards optimal structured CNN pruning via generative adversarial learning," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 2790–2799.

[25] X. Ding, T. Hao, J. Tan, J. Liu, J. Han, Y. Guo, and G. Ding, "ResRep: Lossless CNN pruning via decoupling remembering and forgetting," 2020, arXiv:2007.03260.

[26] Y. Li, S. Lin, B. Zhang, J. Liu, D. Doermann, Y. Wu, F. Huang, and R. Ji, "Exploiting kernel sparsity and entropy for interpretable CNN compression," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 2800–2809.

[27] Y. Zuo, B. Chen, T. Shi, and M. Sun, "Filter pruning without damaging networks capacity," *IEEE Access*, vol. 8, pp. 90924–90930, 2020.

[28] Y. He, P. Liu, Z. Wang, Z. Hu, and Y. Yang, "Filter pruning via geometric median for deep convolutional neural networks acceleration," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 4340–4349.

[29] Yawei Li, Shuhang Gu, Kai Zhang, Luc Van Gool, and Radu Timofte, "DHP: differentiable meta pruning via hypernetworks." In Andrea Vedaldi, Horst Bischof, Thomas Brox, and Jan-Michael Frahm (eds.), *Computer Vision- ECCV 2020- 16th European Conference*, Glasgow, UK, August 23-28, 2020, Proceedings, Part VIII, volume 12353 of Lecture Notes in Computer Science, pp. 608–624.

Appendix

A Supplementary Information

A.1 Hardware Limitations in TEM: Impacts on Deep Learning for Real-Time Processing

The limited memory capacity of Transmission Electron Microscopy (TEM) hardware poses significant challenges for the integration of deep learning methodologies in TEM analysis, primarily due to the demands of real-time image processing applications. Many conventional TEM systems have insufficient network connectivity and data transmission infrastructure, which complicates reliance on external resources for prompt processing. In addition, the user time for TEM equipment is both valuable and limited; consequently, any delays or inefficiencies in data processing can adversely affect productivity and escalate costs, thereby exacerbating the challenges associated with hardware capacity constraints. Furthermore, for organic samples or those sensitive to prolonged or high-intensity beam exposure, delays in real-time TEM image analysis may result in sample degradation, thereby compromising data acquisition. Therefore, this work presents a methodology to mitigate the hardware limitations for transmission electron microscopy.

A.2 Challenges in Processing Transmission Electron Microscopy Datasets Compared to Normal Image Datasets

Transmission electron microscopy datasets, particularly at high resolutions, exhibit significant distinctions from conventional image datasets, primarily attributed to elevated noise levels that can obscure nanoscale details. This noise is generated by various factors, such as camera hardware constraints, electronic interference, fluctuations in sample thickness, sample preparation, and scattering phenomena occurring during transmission.

Furthermore, TEM images necessitate precise calibration and alignment due to their high magnification, in contrast to standard images captured under controlled lighting conditions. The inherent complexity of TEM datasets is further exacerbated by the diverse imaging modes available—such as bright-field, dark-field, and high-angle annular dark field (HAADF)—which complicate segmentation processes. The conditions for capturing TEM data also differ with sample type, which results in the wide variation for TEM images.

In addition to these challenges, TEM datasets are susceptible to artifacts, including those induced by the electron beam, which further complicates the analytical process. Collectively, these factors render the automated processing and segmentation of TEM images particularly challenging, necessitating the application of advanced methodologies to effectively address their unique characteristics.

A.3 Analytical Tasks and Segmentation Complexities in TEM Image Processing

To illustrate the specific challenges encountered in the analysis of TEM images, we present a range of representative tasks:

1. Phase Segmentation: Differentiating and segmenting distinct material phases within TEM images, a fundamental process for revealing the microstructural characteristics of materials. Phases may be intertwined, exhibit transition regions, or possess indistinct boundaries, making it challenging to differentiate them using basic thresholding or contrast-based methods.

2. Defect Detection: Identifying structural imperfections, such as dislocations or voids, that critically influence material properties. The small size of defects and their potential low contrast relative to the surrounding matrix complicate their detection.

3. Feature Recognition: Delineating nanoscale features, such as nanoparticles or thin films, essential for the precise characterization of advanced materials. Variability in imaging conditions and feature type can influence the visibility and contrast of feature recognition, leading to inconsistencies in detection.

4. Quantitative Analysis: Conducting precise quantitative measurements, including feature and phase size distribution, to provide detailed insights into the composition and structure of materials. The accuracy of these measurements depends on the ability to detect and segment TEM images correctly.

Through these examples, we demonstrate the diverse applications of TEM image segmentation and the relevance of our proposed approach in effectively addressing these critical analytical tasks.

B Methodology and Experimental Supplementary Analysis

B.1 Illustrative Explanation of Diversity-Based Two-Phase Pruning Strategy

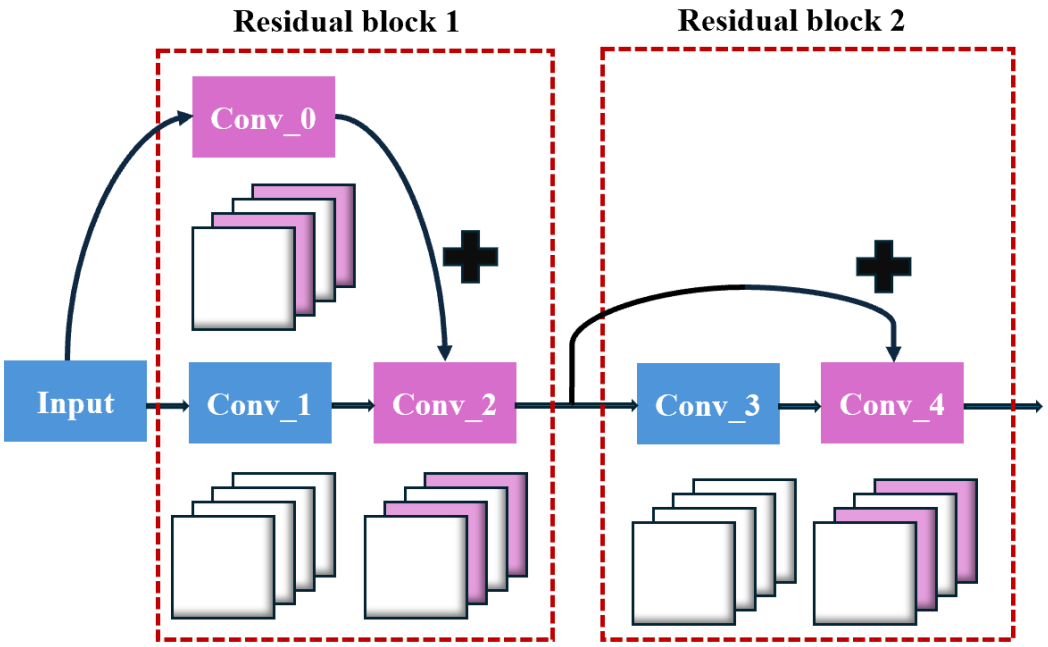

Figure 4: Filters with shortcut connections (e.g., Conv_0, Conv_2, Conv_4) must align to preserve feature maps. A joint evaluation (e.g., Conv_0, Conv_2, Conv_4 together) enhances performance compared to relying solely on the identity projection mapping layer (Conv_0).

In the Residual U-Net architecture, each encoder, decoder, and bottleneck layer contains two residual blocks. When there is a change in the number of channels between blocks, an additional identity projection mapping convolution layer (Conv_0) is introduced to match dimensions. Thus, unlike ResNet-56 or ResNet-110, where shortcut connections directly add zero padding to accommodate increased dimensions, this approach avoids simple, direct addition throughout. Consequently, when pruning, the individual encoder, decoder, and bottleneck layers can be considered independently. Experimental results indicate that, compared to evaluating the identity projection layer with original features in isolation, incorporating the feature extraction information from subsequent residual blocks yields relatively better performance, as illustrated in Fig. 4.

Fig. 5 illustrates the concept of Phase 1: quantitative score pruning. Beyond simply considering the magnitude of filters, additional focus is placed on the diversity of kernels within each filter to maintain generalization in feature extraction. In this figure, each 3x3 kernel can be viewed as a 9-dimensional vector, with each kernel vector having distinct magnitudes and directions. Greater vector diversity enables the extraction of more varied feature values from the feature maps.

Fig. 6 illustrates the concept of Phase 2: Similarity Pruning. In this phase, the average of all kernels within a filter is used to represent the filter as a 9-dimensional vector. If multiple filters yield feature maps that are highly similar, this indicates that the filters serve redundant functions. For instance, the vectors $\vec{F_2}$ and $\vec{F_3}$ in the figure exhibit excessive similarity as determined by Pearson correlation coefficient calculations. During pruning, filters with high similarity are grouped together, retaining only the one with the largest L1-norm magnitude as a representative, while the others are pruned.

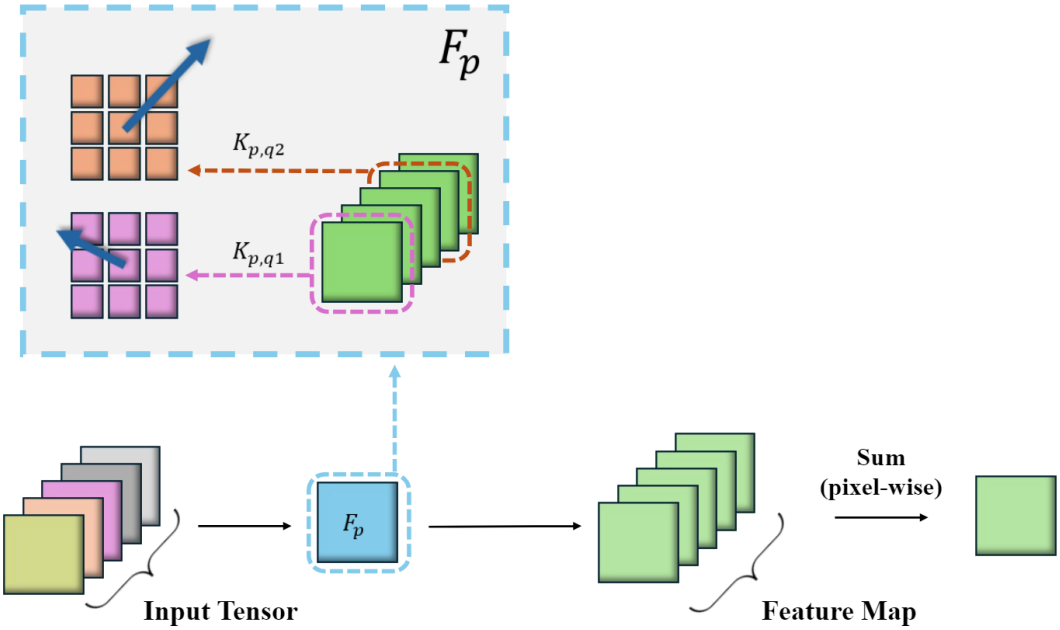

Figure 5: To assess filter significance, we compute the L1-norm of each filter and evaluate kernel diversity by treating 3x3 kernels as 9-D vectors. $K_{p,q}$ represents the q-th kernel within the p-th filter. Diversity is quantified by the variance of both vector lengths and distances of these vectors from the mean vector.

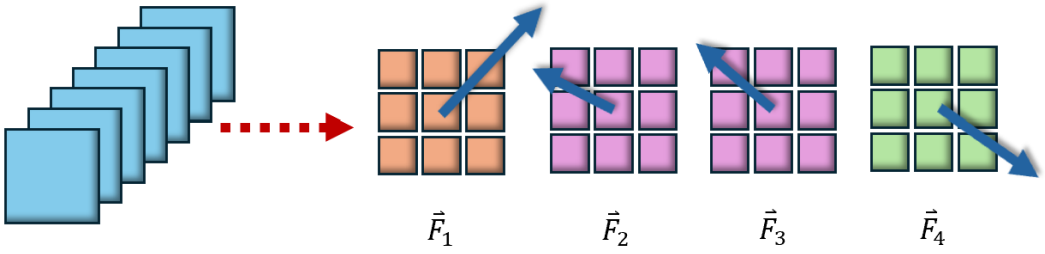

Figure 6: Each filter is represented by the average of its kernels, and the Pearson correlation coefficient is computed between filters. Highly correlated filters (e.g., $\vec{F}_2$ and $\vec{F}_3$ are pruned, keeping only the one with the largest L1-norm.)

B.2 Experimental Setup and Segmentation Results Pre- and Post-Pruning

The channel numbers for the first three encoder layers of the Residual U-Net are 32, 64, and 128, respectively, with corresponding values in the decoder and a bottleneck of 256. During pre-training, the model was trained for 30 epochs using the AdamW optimizer with a fixed learning rate of 5e-5 and a weight decay of learning rate multiplied by 1e-4. The final validation accuracy achieved was 96.33%. After pruning, fine-tuning was conducted for 20 epochs, with learning rates set at 2e-5 for epochs 1 to 6, 1e-5 for epochs 7 to 12, and 1e-6 for epochs 13 to 20. The loss function used during pre-training and fine-tuning of the model is a combination of the Dice coefficient and Binary Cross-entropy. The learning rate during fine-tuning was fixed and decreased over the epochs; thus, the fine-tuning process did not adjust hyperparameters based on the validation set performance. Consequently, the validation set did not serve an optimization role in this training procedure but was solely used to evaluate the model's final performance, effectively functioning as a test set.

In Fig. 7, we present a series of TEM images to evaluate the model's segmentation performance before and after pruning. The pruning criteria was set to 70% pruning in Phase 1 and a Pearson correlation coefficient of 0.8 in Phase 2. From the figure, it is evident that despite the model's size

being significantly reduced from 15.31 MB to 820.69 KB, its segmentation performance remains notable, demonstrating that the reliability of the pruned model is still well-maintained.

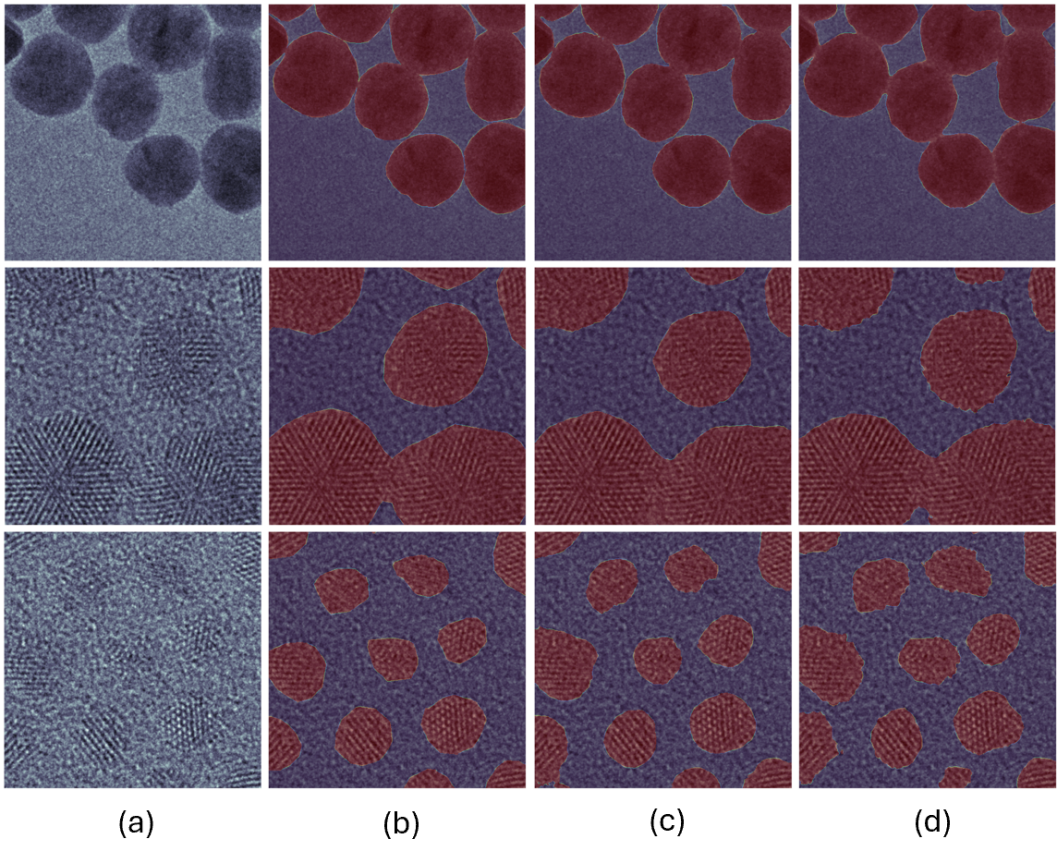

Figure 7: (a) Original TEM image (b) Corresponding ground truth (c) Segmentation results before pruning (d) Segmentation results after pruning.

C Comparison of Variations Across Different Phase 1 and Phase 2 Parameters

In this section, we present the model performance under various Phase 1 and Phase 2 pruning criteria. For Phase 1, we used increments of 10% as pruning thresholds, while for Phase 2, we selected high correlation coefficients $r = 0.7$, $r = 0.8$, and $r = 0.9$. The results of each criterion are shown in a line chart in Fig. 8 and Table 2. It can be observed that at $r = 0.8$ and $r = 0.9$, accuracy generally decreases as the Phase 1 pruning ratio increases. However, at $r = 0.7$, two phenomena emerge: first, pruning 40% of parameters in Phase 1 results in a higher parameter reduction than pruning 50%, indicating that when more parameters are retained in Phase 1, Phase 2 is able to identify and remove more groups of highly similar filters for these two criteria. Secondly, the accuracy for the 50% Phase 1 pruned model is higher than that of the 30% pruned model, suggesting that redundant filters may negatively impact feature discrimination accuracy.

Table 2 presents the model accuracy along with the parameters and their corresponding memory size under different Phase 1 and Phase 2 pruning criteria. The results, highlighted in bold text, represent the case of 70% pruning in Phase 1 and a Pearson correlation coefficient of $r = 0.8$ in Phase 2, achieving a size reduction of 95% with an accuracy drop of less than 1%. Meanwhile, the values in red text for the case of 10% pruning in Phase 1 and a Pearson correlation coefficient of $r = 0.8$ indicate the configuration with accuracy closest to the original model. This configuration has a parameter count of 880,971 (3.36 MB), representing a size reduction of 78.05% compared to the original model, while still maintaining performance levels comparable to those of the original model.

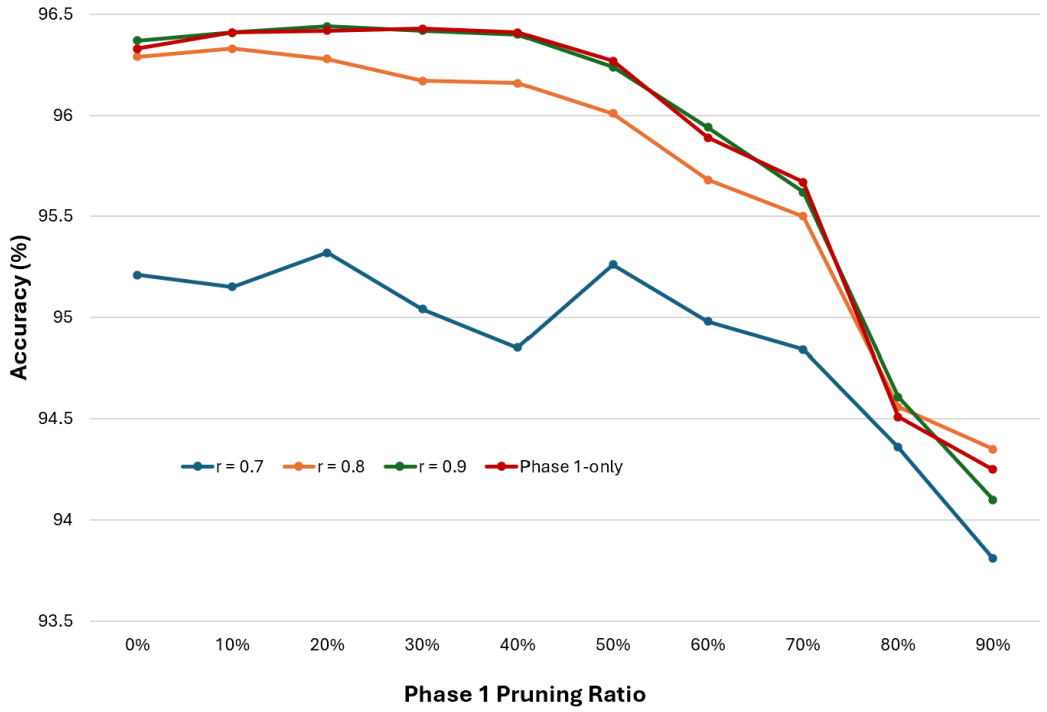

Figure 8: Different pruning ratios in Phase 1 are explored with various Pearson correlation coefficients of $r = 0.7$, $r = 0.8$, and $r = 0.9$ in Phase 2, with values of $r \geq 0.7$ indicating high correlation.

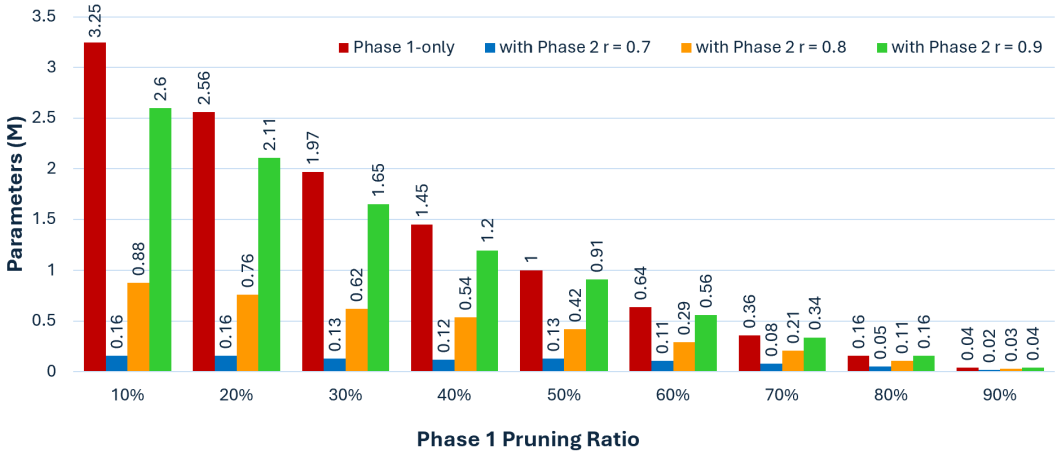

Figure 9: Comparison of Model Size Between Phase 1-Only and Post-Phase 2 Incorporation.

Table 2: Model accuracy and size as a function of Phase 1 pruning ratio (PR) and Phase 2 Pearson coefficient r

	$r = 0.7$	$r = 0.8$	$r = 0.9$
PR	**Original Model Acc. = 96.33% / Parameters = 4012963 (15.31MB)**		
0%	95.21% / 172861(675KB)	96.29% / 977174(3.7MB)	96.37% / 3155064(12.0MB)
10%	95.15% / 158689(620KB)	**96.33% / 880971(3.4MB)**	96.41% / 2595853(9.9MB)
20%	95.32% / 156214(610KB)	96.28% / 758610(2.9MB)	96.44% / 2111431(8.1MB)
30%	95.04% / 129677(507KB)	96.17% / 623998(2.4MB)	96.42% / 1648719(6.3MB)
40%	94.85% / 121289(474KB)	96.16% / 536166(2.1MB)	96.40% / 1202421(4.6MB)
50%	95.26% / 125957(492KB)	96.01% / 418850(1.6MB)	96.24% / 905017(3.5MB)
60%	94.98% / 107688(421KB)	95.68% / 294806(1.1MB)	95.94% / 562465(2.2MB)
70%	94.84% / 80177(313KB)	**95.50% / 210097(821KB)**	95.62% / 337421(1.3MB)
80%	94.36% / 54173(212KB)	94.56% / 107730(421KB)	94.61% / 155121(606KB)
90%	93.81% / 18980(74KB)	94.35% / 31086(121KB)	94.10% / 39628(155KB)

Fig. 9 illustrates the accuracy achieved during the Phase 1-only experiments, as well as the reduction in the number of parameters for the Phase 1-only pruned model following the implementation of Phase 2. It is evident that, at a ratio of $r = 0.8$ in Phase 2, a substantial reduction in parameters can be achieved while maintaining an acceptable level of accuracy compared to the Phase 1-only results.

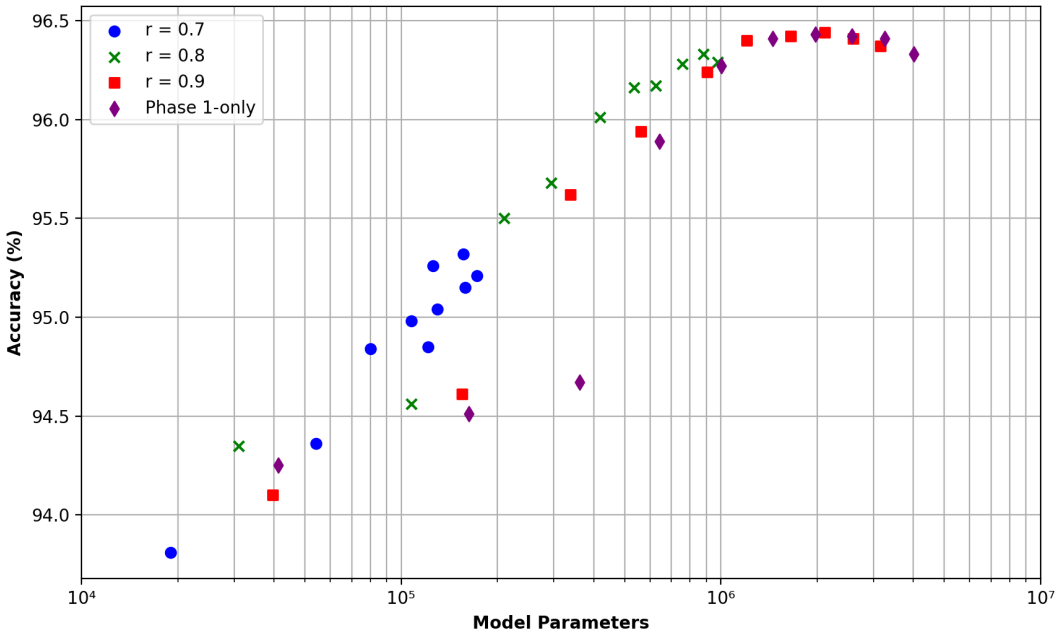

Figure 10: Scatter plot of model parameters, and accuracy across various Phase 1 and Phase 2 criteria. Different r values for Phase 2 are represented by distinct colors, with each color corresponding to parameter counts arranged from high to low, reflecting Phase 1 pruning ratios of 0%, 10%, 20%, up to 90% (except for the case of $r = 0.7$, where the parameter count for Phase 1 with 40% pruning is smaller than that for Phase 1 with 50% pruning).

In Fig. 10, we present scatter plots of model parameters and corresponding fine-tuning accuracy under different criteria selections, illustrating the relationship between changes in model parameters and accuracy. The relationship between the overall parameters and accuracy demonstrates a positive correlation. Notably, the group corresponding to $r = 0.7$ tends to cluster in the lower-left quadrant, indicating structural and accuracy degradation associated with this correlation coefficient. In contrast, the remaining three groups exhibit a more balanced distribution. The scatter trends provide valuable insights for making informed selections of correlation coefficients during the Phase 2 pruning process.

The highest accuracy was attained not with the configuration containing the maximum number of model parameters but with a configuration reduced to approximately 50.88% of the original model's parameter count. This finding suggests that excessive, non-essential weights can inadvertently detract from model performance, highlighting the advantages of targeted pruning strategies. By reducing model parameters to just 10.44% of the original size, we maintain an accuracy exceeding 96%, utilizing a Phase 1 pruning ratio of 50% and a Pearson correlation coefficient of $r = 0.8$. Furthermore, within similar parameter ranges, $r = 0.8$ outperforms $r = 0.9$, while $r = 0.7$ shows variability in performance relative to $r = 0.8$. This inconsistency is attributed to variations in Phase 1 pruning ratios, which impact the retention of essential filters. Despite reducing parameter size by over 100-fold, accuracy only marginally declined from the original 96.33% to 94.35% at a Phase 1 pruning ratio of 90% and a Phase 2 Pearson correlation coefficient of $r = 0.8$. Additionally, the Phase 1-only approach demonstrates suboptimal performance when parameter sizes fall below 10^6, thereby underscoring the necessity of incorporating Phase 2 pruning to enhance overall model efficacy.

D Ablation study

We perform ablation studies on our proposed methods to assess the contribution of the individual algorithmic components we have introduced using a Residual U-Net architecture. To ensure fairness, we implement a 70% pruning in Phase 1, as the outcomes following Phase 2 pruning vary significantly based on the combinations of different components.

In this study, we explore two distinct fine-tuning strategies. Fine-tuning Strategy 1 consists of 1 to 10 epochs with a learning rate of 1e-5, followed by 11 to 20 epochs with a learning rate of 1e-6. Fine-tuning Strategy 2 involves 1 to 6 epochs at a learning rate of 2e-5, 7 to 12 epochs at a learning rate of 1e-5, and 13 to 20 epochs at a learning rate of 1e-6.

Table 3: Ablation study on various combinations of pruning algorithmic components

Method Name	Fine-tuning 1	Fine-tuning 2
Filter Magnitude-only	95.39%	95.61%
Filter Magnitude + var(Kernel Vector Length)	95.16%	95.58%
Filter Magnitude + var(Kernel Vector Distance)	95.14%	95.47%
Proposed Phase 1 Comprehensive Pruning Framework	**95.44%**	**95.70%**

The results presented in Table 3 indicate that among the various fine-tuning approaches, the method that incorporates the filter magnitude as well as differences in length and directional distances of the internal kernels performs the best. This approach demonstrates a comprehensive understanding of the underlying feature extraction mechanisms within the model, as it captures both the overall significance of the filters and the nuanced characteristics of individual kernels. By considering these additional factors, this method enhances the model's ability to retain essential features while discarding redundant information, ultimately leading to improved performance in segmentation tasks.

Furthermore, the findings from the ablation study indicate that pruning strategies should not solely rely on filter magnitudes but should also account for the intricate relationships between kernels. This insight can inform future pruning methodologies and fine-tuning processes, encouraging researchers to explore multi-faceted approaches that leverage both magnitude and kernel diversity. Our findings underscore the importance of a comprehensive approach to model optimization, which may yield significant improvements in performance while reducing model complexity.

E Comparisons

E.1 Residual U-Net Compression: Proposed Pruning Strategy Versus SqueezeNet Fire Module

We compare our proposed pruning approach with the established compression technique employed in SqueezeNet [17]. The fire module in SqueezeNet has been shown to significantly reduce the parameter count in convolutional layers and has recently been adapted for U-Net architectures [18, 19]. For convolutional layers with a 3x3 kernel size, the fire module replaces the conventional

structure with a combination of squeeze and expand layers. The squeeze layer utilizes a 1x1 kernel and reduces the number of output channels to one-fourth of the original, while the expand layer is divided into two parts, utilizing kernel sizes of 1x1 and 3x3, each producing output channels at half the original count.

We applied the principles of the fire module to all convolutional layers with a kernel size of 3x3 in the original residual U-Net architecture. This adaptation reduced the model's parameter count from the original 4,012,963 (15.31 MB) to 766,027 (2.92 MB). Consequently, we trained the squeeze residual U-Net for 30 epochs, starting with an initial learning rate of 1e-4, which was decreased by a factor of 0.1 every 10 epochs to facilitate improved convergence due to the significant reduction in parameters, while also aligning the final learning rate with that used in the original fine-tuning experiments.

Table 4: Performance of the squeeze residual U-Net compared to the diversity-based two-phase pruned model

	Accuracy	**Parameters**	**MACs**	**Prediction(GPU)**	**Prediction(CPU)**
Original	96.33%	4012963	$\approx 79G$	16.27ms / image	347.63ms / image
SqueezeNet	95.97%	766027	$\approx 14.35G$	16.25ms / image	306.21ms / image
Reduction	0.36%	80.91%	**84.84%**	0.12%	11.91%
Ours	96.28%	758610	$\approx 27.69G$	12.38ms / image	170.86ms / image
Reduction	**0.05%**	**81.10%**	64.95%	**23.91%**	**50.85%**

The comparative results are presented in Table 4. To facilitate a comparison with the squeezed model, we selected the pruning model based on Phase 1 with 20% pruning and Phase 2 utilizing a Pearson correlation coefficient of 0.8, as it yields a parameter count of 758,610 (2.89 MB), which is closest to that of the squeezed model. The two-phase pruning strategy demonstrates superior performance across all evaluated metrics except for one (MAC reduction), particularly excelling in prediction time. Notably, for the squeeze model, despite the significant reduction in the number of parameters and MACs, its prediction time did not exhibit a significant decrease and remained nearly equivalent to that of the original model. However, our methodology exhibited a 23.91% reduction in GPU prediction and 50.85% reduction in CPU prediction. This demonstrates that our proposed strategy not only effectively reduces parameter count while maintaining accuracy but also significantly accelerates real-time image processing.

E.2 Implementation of Two-Phase Pruning Strategy to ResNet-56 and ResNet-110 on CIFAR-10

To validate the generalizability of the proposed pruning strategy across various model architectures and tasks, and to compare it with other state-of-the-art structured pruning methods, we applied the pruning approach to the CIFAR-10 dataset using ResNet-56 and ResNet-110. Unlike the previously utilized residual U-Net architecture, ResNet-56 and ResNet-110 do not incorporate identity projection mapping convolution layers when there is a change in the number of channels. Instead, their shortcut connections directly add zero padding to accommodate increased dimensions, resulting in a configuration where layers associated with shortcut connections are linked end-to-end. The distance between the initial and final layers is considerable, with significant differences in feature extraction, and the indices of the shortcut connection layers must remain consistent. Therefore, in this evaluation, we focused solely on pruning the first convolution layer within the residual blocks that are independent of the shortcut connections.

The pre-training and fine-tuning strategies were consistent, employing a total of 165 epochs. We utilized stochastic gradient descent (SGD) as the optimizer, with a momentum of 0.9. The initial learning rate was set at 0.01, which was increased tenfold after 500 steps, followed by a reduction by a factor of 0.1 at 32,000 and 48,000 steps, respectively. The accuracy of the pre-trained ResNet-56 was 92.87%, while ResNet-110 achieved an accuracy of 93.40%, serving as a baseline for evaluating the two-phase pruning strategy.

All data pertaining to SOTA pruning methods presented in Table 5 are derived from their respective original publications or third-party replicated publication. In comparing the pruned versions of ResNet-56 and ResNet-110, we selected a pruning criterion of 20% in Phase 1, along with a Pearson

Table 5: Performance comparison of ResNet-56 and ResNet-110 on the CIFAR-10 dataset

Method Name	Original Acc.	Pruned Acc.	Acc. Diff.	Params. Reduction
ResNet-56 on CIFAR-10 Ours: Phase 1 20% pruned; Phase 2 correlation threshold $r = 0.8$				
L1-norm pruned-A [6]	93.04%	93.10%	-0.06%	9.4%
L1-norm pruned-B [6]	93.04%	93.06%	+0.02%	13.7%
RFP [8]	93.39%	93.12%	-0.27%	23.7%
ABCPruner [20]	93.26%	93.23%	-0.03%	54.2%
TMI-GKP [21]	93.78%	94.00%	+0.22%	43.5%
HRank [22]	93.26%	93.17%	-0.09%	54.2%
OED [23]	93.97%	92.29%	-1.68%	60.0%
GAL [24]	93.97%	91.58%	-2.39%	65.9%
ResRep [25]	93.71%	93.71%	0.00%	-
KSE [26]	93.03%	92.88%	-0.15%	57.6%
FPGM [28]	93.59%	93.26%	-0.33%	-
DHP [29]	92.95%	92.94%	-0.01%	58.9%
Ours	92.87%	92.90%	+0.03%	**67.3%**
ResNet-110 on CIFAR-10 Ours: Phase 1 20% pruned; Phase 2 correlation threshold $r = 0.8$				
L1-norm pruned-A [6]	93.53%	93.55%	-0.06%	2.3%
L1-norm pruned-B [6]	93.53%	93.30%	+0.02%	32.4%
RFP [8]	93.65%	93.27%	-0.38%	34.2%
ABCPruner [20]	93.50%	93.58%	+0.08%	67.4%
TMI-GKP [21]	94.26%	94.90%	+0.64%	43.5%
HRank [22]	93.50%	93.36%	-0.14%	59.2%
OED [23]	94.10%	93.60%	-0.50%	48.8%
GAL [24]	94.10%	92.74%	-1.36%	44.8%
ResRep [25]	94.64%	94.62%	-0.02%	-
FPDNC [27]	93.82%	93.78%	-0.04%	48.2%
FPGM [28]	93.68%	93.74%	+0.06%	-
DHP [29]	94.69%	94.63%	-0.06%	36.8%
Ours	93.40%	93.42%	+0.02%	**68.2%**

correlation coefficient $r = 0.8$ in Phase 2, aiming for an accuracy that closely matches the original model. For ResNet-56, the original accuracy was 92.87%. After pruning, despite a substantial reduction of 67.3% in the number of parameters, the accuracy did not decline; instead, it improved by 0.03%, reaching 92.90%.

For ResNet-110, the original accuracy was 93.40%. After pruning, the number of parameters was significantly reduced by 68.2%, and the accuracy improved by 0.02%, reaching 93.42%. Among the pruning methods, TMI-GKP exhibited the best accuracy post-pruning, with a parameter reduction amounting to approximately only two-thirds of that achieved by our method. By solely pruning the first layer of the residual block and significantly reducing the number of parameters, our two-phase pruning strategy still achieved comparable performance with ResNet-56 and ResNet-110, even slightly outperforming the original models. The results demonstrate the efficacy of our pruning strategy in maintaining or improving model accuracy while substantially reducing parameter count, displaying its potential for further optimization in future applications. Furthermore, this research paves the way for exploring the adaptability of our strategy across various architectures and tasks, potentially facilitating broader applications in resource-constrained environments where model size and computational efficiency are critical.