We Need Far Fewer Unique Filters Than We Thought

Zahra Babaiee
TU Vienna & MIT
zbabaiee@mit.edu

Peyman M. Kiasari
TU Vienna
peyman.kiasari@tuwien.ac.at

Daniela Rus
MIT
rus@mit.edu

Radu Grosu
TU Vienna
radu.grosu@uwien.ac.at

Abstract

We challenge the conventional belief that CNNs require numerous distinct kernels for effective image classification. Our study on depthwise separable CNNs (DS-CNNs) reveals that a drastically reduced set of unique filters can maintain performance. Replacing thousands of trained filters in ConvNextv2 with the closest linear transform from a small filter set, results in small accuracy drops. Remarkably, initializing depthwise filters with **only 8 unique frozen filters**, achieves minimal accuracy drop on ImageNet. Our findings question the necessity of numerous filters in DS-CNNs, offering insights into more efficient network designs.

1 Introduction

Convolutional Neural Networks (CNNs) have revolutionized computer vision, and as CNNs scaled to millions of parameters [25, 8, 12], Depthwise Separable CNNs (DS-CNNs) emerged as an efficient variant [11, 10]. State-of-the-art architectures like ConvNeXt [21] utilize up to 50,000 trainable spatial filters. Recent studies have revealed inherent repeating patterns in these filters, with Trockman et al.[28] investigating their covariance structure and Babaiee et al.[3][4] has shown that these filters exhibit highly clusterable patterns across various architectures. Intriguingly, these patterns can be classified into a few categories related to the Difference of Gaussians (DoG) functions. These findings raise an important question:

Is training thousands of filters necessary if similar patterns appear across layers?

In this paper, we challenge the conventional belief that a large number of unique filters is essential for CNNs. Instead, we explore the potential of using a limited set of carefully chosen filters. We demonstrate that through a greedy search, we progressively reduce the number of unique filters and discover that the accuracy remains relatively unchanged even with only 8 filters. Remarkably, these 8 filters bear a striking resemblance to DoGs, Gaussians, and the first derivatives of Gaussians, which are well-established in scale-space theory [17] and mammalian vision modeling [29, 30]. We validate the effectiveness of these discovered filters on other datasets and model architectures, consistently observing their comparable performance, especially in scenarios with limited training data.

Our findings challenge the prevailing notion that large filter variety is necessary in CNNs. In summary, the key contributions of our work are:

- We show that replacing each ConvNeXt filter with its the closest linear transformation chosen from 8 filters closely approximates the original model's performance **without fine-tuning**.
- Moreover, training the model with only **8 filters** + bias reasonably preserves accuracy.
- Our findings challenge the conventional belief that a large number of unique filters is essential for CNNs, by reducing the number of unique filters to only 8.

Workshop on Scientific Methods for Understanding Deep Learning, NeurIPS 2024.

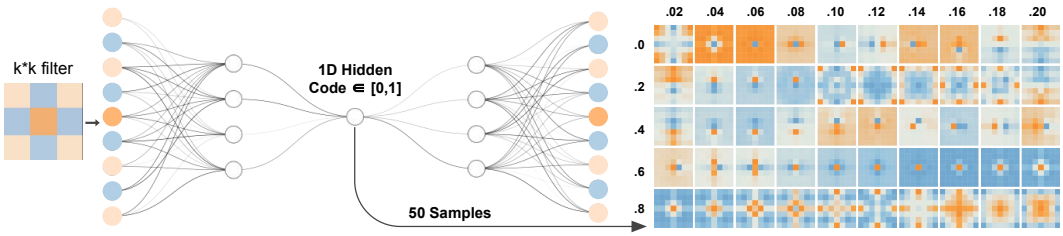

Figure 1: We uniformly sample candidates from the code layer of the filter Autoencoder, which we have trained on the filters that have been learned by the models.

2 Related Work

Depthwise Convolutions (DCs) have revolutionized Convolutional Neural Networks (CNNs), leading to efficient architectures like MobileNet [11], EfficientNet [26], and ConvNeXt [21]. Recent studies have revealed structured covariance matrices in large-kernel DCs [28] and the convergence of depthwise convolutional kernels into clusters resembling Gaussian derivatives [3][4]. While filter pruning techniques have been explored to reduce computational complexity [9, 16], our work investigates filter diversity in DS-CNNs, challenging the need for numerous unique filters. We establish a connection with scale-space theory [13, 17], which examines signals across scales using Gaussian derivatives. Lindeberg's computational framework for visual receptive fields [17, 20] aligns with hierarchical stages in mammalian visual systems [18] and provides a provable approach to capture image transformations [19], further supported by center-surround receptive field models that enhance robust image classification [2]. Our findings show that a small set of carefully chosen filters, including Gaussians, Difference of Gaussians (DoG), and their derivatives, can effectively replace numerous learned filters in DS-CNNs, aligning with scale-space theory principles and suggesting limited inherent diversity in learned depthwise filters.

3 Do We Require Thousands of Distinct Filters?

In this section, we investigate whether employing thousands of unique filters is essential for maintaining the performance of DS-CNNs. In particular, we explore what is the impact on the performance of the network, when we replace the trained filters with a minimal set of distinct filter variations.

3.1 The Quest for Optimal Filters

In order to explore the possibility of reducing the number of distinct filters in DS-CNNs, we sought to distill the filters of trained models into a compact set. We collected filters from ConvNeXtv1 and ConvNeXtv2 models of various sizes and employed an autoencoder to learn a compressed representation of these filters. The autoencoder was trained to encode each filter into a single dimension, following a similar procedure as the one described in [3].

We collected and normalized 7×7 depthwise filters from all layers of our model bank networks. The autoencoder architecture consists of an encoder and decoder, each with four intermediate layers. The encoder uses Leaky ReLU activations and a sigmoid in the code layer, mapping to [0, 1]. The decoder mirrors this structure, ending with a tanh activation to reconstruct normalized filters within [-1, 1]. This architecture enables learning compact representations of the filter space, facilitating exploration of filter variations and potential reduction in distinct filters for depthwise separable CNNs.

We uniformly sampled 100, 50, 25, and 10 points from the autoencoder's code layer to generate distinct filter sets. For each set of samples, separately, we replace each original filter of the model with the closest linear transformation of a single, best-matching filter from the set. (see Appendix A.1). Table 3 shows the accuracy of ConvNeXtV2 models before and after filter replacement. Notably, approximating all filters with just 100 sampled filters maintains robust performance *without fine-tuning*, especially for larger models. For ConvNeXt V2 Huge, replacing nearly 50K filters with 100 sampled filters results in less than 2% accuracy drop.

As expected, reducing the number of sampled filters decreased accuracy, especially with only 10 filters. To achieve a better smaller set, we conducted a systematic greedy search on ConvNeXt-v2-Tiny using 50 uniformly sampled filters, removing the least important filters one by one. Figure 1 illustrates the 50 uniformly sampled samples. The accuracy plot (Figure 2) showed stability until the

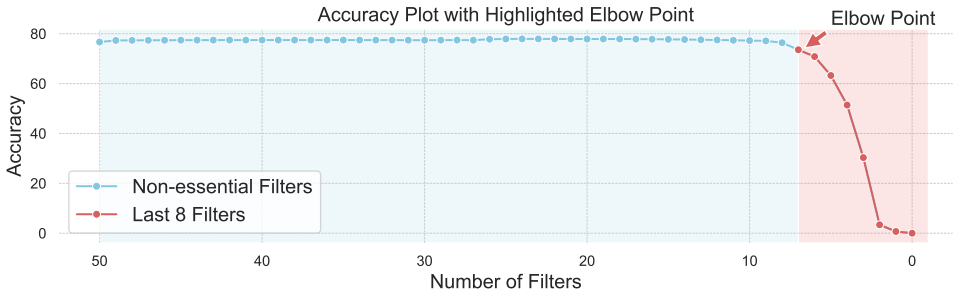

Figure 2: Our systematic greedy search for the essential filters in ConvNeXt-v2-Tiny and ConvNeXt-v2-Pico. While the removal of most filters did not noticeably changed accuracy, 8 of them were essential.

removal of the last 8 filters, wich is the elbow point. These 8 filters outperformed the 25-filter set across ConvNeXt V2 models, effectively replacing up to 50K filters while maintaining acceptable performance. (The last row of Table 3) To validate generalizability, we applied this approach to ConvNeXt-v2-Pico and Large, arriving at a similar set of 8 filters. This consistency suggests the existence of universal filters capturing fundamental patterns across model architectures.

3.2 Understanding the Eight Filters

This subsection analyzes the functional characteristics of the eight filters (Figure 3) identified through greedy search, examining their resemblance to traditional image-processing operators and their roles in network feature extraction.

Filters 1-4: These 4 filters resemble central difference operators, approximating Gaussian derivatives discretely, akin to edge detection and texture analysis models.

Filters 5-6: These 2 filters resemble 1st order Gaussian derivatives along x and y axes. Their pronounced spatial smoothing captures broader textural information, potentially improving generalization across visual contexts.

Filters 7: This filter resembles a 2-D discrete Difference of Gaussians (DoG), approximating the Laplacian of Gaussian. Crucial for blob detection and bar pattern recognition, it likely enhances the model's edge and contour detection capabilities.

Filter 8: This filter resembles a fine-scaled Gaussian kernel, used for noise reduction while preserving edges. Gaussian filters are uniquely proven for scale-space representation in image processing.

Filters formal definition: For completeness, we provide below the formal definition of the continuous functions corresponding to the 2D Gaussian, the 2D derivative of the Gaussian along the x and the y axis, respectively, and the 2D difference of Gaussians (DoG, Laplacian, Mexican hat):

$$\begin{aligned} \text{(Gaussian)} \quad & G(x,y) && = e^{-(x^2+y^2)/2\sigma^2}/2\pi\sigma^2 \\ \text{(dGaussian/dx)} \quad & \partial G/\partial x && = -xG(x,y)/\sigma^2 \\ (\Delta\text{Gaussian}) \quad & \text{DoG}(x,y) && = e^{-(x^2+y^2)/2\sigma_1^2}/2\pi\sigma_1^2 - e^{-(x^2+y^2)/2\sigma_2^2}/2\pi\sigma_2^2 \end{aligned}$$

4 Experiments

In this section, we evaluate model performance on ImageNet using the eight identified filters, through fine-tuning with closest linear transformation and training from scratch with frozen filters. Results show negligible accuracy drops.

4.1 ImageNet

Table 3 shows model accuracy remains stable despite reduced filter diversity, prompting further investigation. Fine-tuning with frozen filters (Table 2) yields minimal accuracy drops for Pico (-1%) and Tiny (-0.4%) models. In these model filters are linear transfer of one of the 8 filters, in mathematical expressions they are $a(x+b)$. Given DS-CNNs architecture, coefficient can be

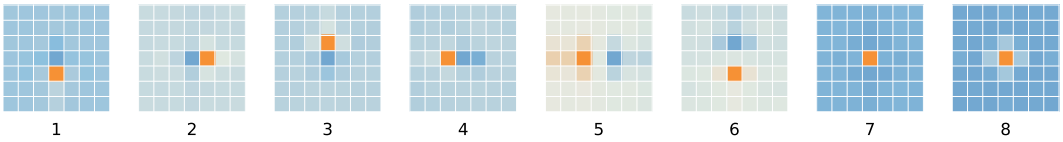

Figure 3: Heatmap of the 8 filters found through greedy search on ConvNeXtv2 tiny model.

transferred to the fully-connected layers, simplifying the filters essentially to $x + b$, where b acts as bias. This motivated us to train a model from scratch using only these 8 filters plus biases.

Training with Only 8 Frozen Transferred Filters plus Bias. To further investigate the effectiveness of the 8 candidate filters, we trained ConvNeXtv2 models from scratch, initializing each layer's filters with the transferred 8 filters with trainable bios. We trained the networks for 300 epochs following the same training pipeline as described in the original paper [24], but with frozen filters having a trainable bios.

Table 1 shows the results. Remarkably, the ConvNeXtv2 Tiny model with only 8 types of filters achieved an accuracy of 82.7%, which is only 0.2% lower than the model trained with 6,624 trainable filters and FCMAE pretraining. We also conducted a similar experiment with the smaller ConvNeXtv2 Pico model, where the model with 8 types of frozen filters reached an accuracy of 80.2%, 0.1% lower than the model with 2,944 trainable filters.

Table 1: Imagenet Top-1 Accuracy. We report accuracies for original models both with and without FCMAE pretraining. It is quite remarkable that the model with only 8 unique filters achieve comparable results.

ConvNeXtv2 Models	**Pico**	**Tiny**	**Base**	**Large**
Number of Original Filters	2 944	6 624	18 048	27 072
Original model with FCMAE[1]	80.3%	82.9%	84.9%	85.8%
Original model	79.7%	82.5%	84.3%	84.5%
With 8 unique filters + bias	80.2%	82.7%	84.6%	85.4%

[1] FCMAE (fully convolutional masked autoencoder framework) is a heavy pretraining ConvNextv2 uses.

These results demonstrate the effectiveness of the candidate filter set in capturing the essential features required for the task, even when the filters are frozen and not learned during training. The ability to achieve competitive performance with only 8 fixed filter types highlights the potential for using a small set of carefully selected filters and challenges the belief of the high diversity of filters required.

By training the models from scratch with frozen filters, we eliminate the need for learning the filter weights during training, which can reduce the computational complexity and memory requirements of the training process, especially in larger models with larger kernel sizes.

4.2 Conclusion

In this paper, we demontrated that DS-CNNs perform with a reasonably high accuracy with only 8 unique, not-trained filters. We thus questioned the necessitiy of training thousands of distinct filters in CNNs (Each DS-CNN, is a CNN).

In this work, we searched for the 8 filters through a compressed filter set derived from the trained models. Future research could explore training models with a small number of shared filters across all layers, potentially discovering even more effective filter sets. Our results open new avenues for designing novel CNN architectures, enhancing model efficiency, and improving explainability in CNNs.

References

[1] Diversity of filters within a conv layer. `https://stackoverflow.com/questions/50858028/caffe-cnn-diversity-of-filters-within-a-conv-layer`, 2018. Accessed: 2024.

[2] Zahra Babaiee, Ramin Hasani, Mathias Lechner, Daniela Rus, and Radu Grosu. On-off center-surround receptive fields for accurate and robust image classification. In Marina Meila and Tong Zhang, editors, *Proceedings of the 38th International Conference on Machine Learning*, volume 139 of *Proceedings of Machine Learning Research*, pages 478–489. PMLR, 18–24 Jul 2021.

[3] Zahra Babaiee, Peyman Kiasari, Daniela Rus, and Radu Grosu. Unveiling the unseen: Identifiable clusters in trained depthwise convolutional kernels. In *The Twelfth International Conference on Learning Representations*, 2023.

[4] Zahra Babaiee, Peyman M. Kiasari, Daniela Rus, and Radu Grosu. Neural echos: Depthwise convolutional filters replicate biological receptive fields. In *Proceedings of the IEEE Winter Conference on Applications of Computer Vision (WACV)*, pages 8201–8210, 2024.

[5] Adam Coates, Andrew Y. Ng, and Honglak Lee. An analysis of single-layer networks in unsupervised feature learning. *Proceedings of the fourteenth international conference on artificial intelligence and statistics*, pages 215–223, 2011.

[6] Paul Gavrikov and Janis Keuper. Cnn filter db: An empirical investigation of trained convolutional filters. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 19066–19076, June 2022.

[7] Casey A. Graff and Jeffrey Ellen. Correlating filter diversity with convolutional neural network accuracy. In *2016 15th IEEE International Conference on Machine Learning and Applications (ICMLA)*, pages 75–80, 2016.

[8] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep Residual Learning for Image Recognition. In *Proceedings of 2016 IEEE Conference on Computer Vision and Pattern Recognition*, CVPR '16, pages 770–778. IEEE, June 2016.

[9] Torsten Hoefler, Dan Alistarh, Tal Ben-Nun, Nikoli Dryden, and Alexandra Peste. Sparsity in deep learning: Pruning and growth for efficient inference and training in neural networks, 2021.

[10] Andrew Howard, Mark Sandler, Grace Chu, Liang-Chieh Chen, Bo Chen, Mingxing Tan, Weijun Wang, Yukun Zhu, Ruoming Pang, Vijay Vasudevan, Quoc V. Le, and Hartwig Adam. Searching for mobilenetv3. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, October 2019.

[11] Andrew G. Howard, Menglong Zhu, Bo Chen, Dmitry Kalenichenko, Weijun Wang, Tobias Weyand, Marco Andreetto, and Hartwig Adam. Mobilenets: Efficient convolutional neural networks for mobile vision applications. *CoRR*, abs/1704.04861, 2017.

[12] Gao Huang, Zhuang Liu, Laurens van der Maaten, and Kilian Q. Weinberger. Densely connected convolutional networks. In *CVPR*, pages 2261–2269. IEEE Computer Society, 2017.

[13] Jan J Koenderink. The structure of images. *Biological cybernetics*, 50(5):363–370, 1984.

[14] Alex Krizhevsky, Vinod Nair, and Geoffrey Hinton. Cifar-10 (canadian institute for advanced research).

[15] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. In F. Pereira, C. J. C. Burges, L. Bottou, and K. Q. Weinberger, editors, *Advances in Neural Information Processing Systems*, volume 25. Curran Associates, Inc., 2012.

[16] Lucas Liebenwein, Cenk Baykal, Harry Lang, Dan Feldman, and Daniela Rus. Provable filter pruning for efficient neural networks, 2020.

[17] Tony Lindeberg. *Scale-space theory in computer vision*, volume 256. Springer Science & Business Media, 2013.

[18] Tony Lindeberg. Normative theory of visual receptive fields. *Heliyon*, 7(1):e05897, 2021.

[19] Tony Lindeberg. Covariance properties under natural image transformations for the generalised gaussian derivative model for visual receptive fields. *Frontiers in Computational Neuroscience*, 17, June 2023.

[20] Tony Lindeberg. Approximation properties relative to continuous scale space for hybrid discretizations of gaussian derivative operators, 2024.

[21] Zhuang Liu, Hanzi Mao, Chao-Yuan Wu, Christoph Feichtenhofer, Trevor Darrell, and Saining Xie. A convnet for the 2020s. *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022.

[22] Maria-Elena Nilsback and Andrew Zisserman. Automated flower classification over a large number of classes. In *2008 Sixth Indian Conference on Computer Vision, Graphics Image Processing*, pages 722–729, 2008.

[23] Omkar M. Parkhi, Andrea Vedaldi, Andrew Zisserman, and C. V. Jawahar. Cats and dogs. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 3498–3505, 2012.

[24] Ronghang Hu Xinlei Chen Zhuang Liu In So Kweon Sanghyun Woo, Shoubhik Debnath and Saining Xie. Convnext v2: Co-designing and scaling convnets with masked autoencoders. *arXiv preprint arXiv:2301.00808*, 2023.

[25] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *CoRR*, abs/1409.1556, 2014.

[26] Mingxing Tan and Quoc V. Le. Efficientnet: Rethinking model scaling for convolutional neural networks. *CoRR*, abs/1905.11946, 2019.

[27] Asher Trockman and J. Zico Kolter. Patches are all you need? *CoRR*, abs/2201.09792, 2022.

[28] Asher Trockman, Devin Willmott, and J Zico Kolter. Understanding the covariance structure of convolutional filters. In *The Eleventh International Conference on Learning Representations*, 2023.

[29] R.A. Young, R.M. Lesperance, and W.W. Meyer. The gaussian derivative model for spatial-temporal vision: I. cortical model. *Spatial vision*, 14(3-4):261–319, 2001.

[30] Richard A. Young. The gaussian derivative model for spatial vision: I. retinal mechanisms. *Spatial Vision*, 2(4):273 – 293, 1987.

[31] Dejun Zhang, Linchao He, Mengting Luo, Zhanya Xu, and Fazhi He. Weight asynchronous update: Improving the diversity of filters in a deep convolutional network. *Computational Visual Media*, 6(4):455–466, December 2020.

A Appendix / supplemental material

A.1 Closest Linear Transformation

A depthwise filter for the c-th channel can be denoted as F_c, where c is the index of the channel. When flattened, F_c can be represented as a vector $\mathbf{f}_c \in \mathbb{R}^{k^2}$, where $k \times k$ is the spatial dimension of the filter. The matrix F composed of these flattened vectors is $F = [\mathbf{f}_1, \mathbf{f}_2, \ldots, \mathbf{f}_C]^T \in \mathbb{R}^{C \times k^2}$.

For each depthwise filter F_c learned by the ConvNeXtV2-tiny model, we then conducted a linear approximation with respect to the decoded filters, by identifying the scalar coefficients a and b, which minimized the Euclidean distance between the corresponding flattened filter vector f_c, and the linear combination $af'_c + b$, where f'_c represents a flattened decoded-filter sample. The original filter was then substituted with the optimal linear combination $af'_c + b$ that exhibited the smallest distance to the original, thus preserving the filter's functional characteristics while reducing model complexity.

In mathematical expressions, given smaller set of fitters G, For each $\mathbf{f}_c \in F$, we want to replace it with the closest linear transformation $a^*\mathbf{g}^* + b^*$, where

$$(a^*, b^*, \mathbf{g}^*) = \underset{a,\, b \in \mathbb{R},\, \mathbf{g} \in G}{\operatorname{argmin}} \|\mathbf{f}_c - (a\mathbf{g} + b)\|.$$

To solve for scalars a and b that minimize the distance between vectors f_c and $af'_c + b$, we use linear regression. Here, the goal is to determine the coefficients a and b for two vectors x and y such that by having $\tilde{y} = ax + b$ the length of the vector $y - \tilde{y}$ is minimized. This problem has a well-known solution.

$$a = \frac{n\sum_{i=1}^n x_i y_i - \sum_{i=1}^n x_i \sum_{i=1}^n y_i}{n\sum_{i=1}^n x_i^2 - (\sum_{i=1}^n x_i)^2} \quad b = \frac{\sum_{i=1}^n y_i \sum_{i=1}^n x_i^2 - \sum_{i=1}^n x_i \sum_{i=1}^n x_i y_i}{n\sum_{i=1}^n x_i^2 - (\sum_{i=1}^n x_i)^2} \tag{1}$$

Calculating Equations (1) can be computationally intensive, especially when dealing with hundreds of thousands of filters. To reduce computational complexity, we can use a normalization trick. Since any linear transformation of x does not alter the optimal $\tilde{y}$, we normalize x using the transformation $\hat{x} = \frac{x - \bar{x}}{||x - \bar{x}||}$. With this normalization, $\sum_{i=1}^n \hat{x}_i = 0$ and $\sum_{i=1}^n \hat{x}_i^2 = 1$, allowing us to simplify Equation(1).

$$a = \frac{n\sum_{i=1}^n x_i y_i}{n} = \langle x, y \rangle \quad b = \frac{\sum_{i=1}^n y_i}{n} = \bar{y} \tag{2}$$

Consequentially, Given the vectors $\hat{x}_1, \hat{x}_2, \ldots, \hat{x}_n$ as the rows of matrix $\hat{X}$ and the vectors $y_1, y_2, \ldots, y_m$ as the columns of matrix Y, we introduce the vector y_{mean}, which contains the means $\bar{y}_1, \bar{y}_2, \ldots, \bar{y}_m$. Using these, we can calculate the coefficients a_{ij} and b_{ij} for each pair of x_i and y_j through matrix multiplication.

$$A = \hat{X}Y \quad B = y_{\text{mean}}\mathbf{1}^\top \tag{3}$$

For each layer, with the set of depthwise filter vectors matrix F and the sample filter vectors matrix F', we calculate the coefficients as above to find the closest linear approximation.

A.2 Fine-tuning

Here, we fine-tuned Pico and Tiny models from the "Acc with 8" row of Table 3 while keeping the filters frozen.

A.3 Other Datasets

To investigate the generalizability of our findings, we extend our experiments to other datasets and compare the performance of the ConvNeXt Femto model across various settings.

Table 2: Demonstrating fine-tuned accuracy of ConVextV2 models with **frozen filters** approximated with the closest linear transformation of the eight filters.

Models	Original Acc	lin. trans. 8 filters	lin. trans. 8 filters + fine-tuning
ConvNeXtv2 Pico	80.3%	73.1%	79.3%
ConvNeXtv2 Tiny	82.9%	76.7%	82.5%

Datasets. We evaluate the low filter variety on five datasets: CIFAR-10 [14], Flowers [22], Pets [23], and STL-10 [5]. These datasets have smaller scales compared to ImageNet, with the size of training sets ranging from 2040 to 50000 samples.

Settings. We use the ConvNeXt Femto model as our base architecture. For a fair comparison, we train the model on all datasets for 300 epochs, following the training parameters from the ConvNeXt paper [21], and keep the training settings consistent across all runs and datasets.

For each dataset, we first train the model to obtain the baseline accuracy. Then, we explore two different filter initialization strategies:

1. Depthwise filters directly transferred from the ConvNeXt Femto model trained on ImageNet.
2. 8 filter types, frozen from scratch.

Table 4 presents the accuracy of the ConvNeXt Femto model on each dataset under these different initialization settings.

Table 3: The accuracy with **no fine-tuning** of the various ConvNeXtv2 models, when their filters are replaced with the closest linearly transformation from uniformly-sampled candidate-filter sets. It is quite remarkable that only 8 filters from our greedy search perform encouragingly well.

ConvNeXtv2 Models	**Pico**	**Tiny**	**Base**	**Large**	**Huge**
Number of Filters	2 944	6 624	18 048	27 072	49 632
Original Acc	80.3%	83.0%	84.9%	85.8%	86.3%
Acc with 100 filters	75.3%	77.0%	80.6%	83.7%	84.6%
Acc with 50 filters	75.0%	75.4%	80.5%	83.2%	84.0%
Acc with 25 filters	72.0%	66.9%	72.8%	79.6%	80.4%
Acc with 10 filters	23.4%	1.0%	1.4%	3.0%	2.0%
Acc with 8 (Greedy Search)	73.1%	76.7%	79.3%	81.2%	82.8%

Results. The results demonstrate the effectiveness of using the 8 frozen filters across different datasets. Notably, the performance improvement becomes more pronounced as the dataset size decreases. For the Flowers and Pets datasets, the frozen 8 filter types achieve remarkable improvements of up to 11% and 30%, respectively, compared to the baseline model.

Interestingly, on these smaller datasets, the frozen 8 filter types even outperform the transferred filters from the model trained on ImageNet. This observation suggests that the carefully selected filter types capture fundamental patterns that are highly relevant to the task at hand, even when the dataset size is limited.

The superior performance of the frozen filters on smaller datasets highlights their ability to extract meaningful features without the need for extensive fine-tuning. This finding has significant implications for scenarios where training data is scarce or computational resources are limited.

Table 4: Comparing accuracy of ConvNeXt Femto model with different training setups. Demonstrating the superiority of using **8 (3) unique frozen filters** to even transferred filters of pretrained ImageNet for pets and flowers datasets.

Dataset	CIFAR10	STL-10	Oxford Flowers	Oxford Pets
# Training Set Size	50000	5000	2040	3680
Original Acc	96.9%	80.4%	66.0%	36.3%
Acc with ImageNet Filters	97.1%	83.2%	73.2%	56.0%
Acc with 8 unique Filters	96.3%	83.1%	77.7%	66.4%

B Debunking Challenge Submission

B.1 What commonly-held position or belief are you challenging?

Provide a short summary of the body of work challenged by your results. Good summaries should outline the state of the literature and be reasonable, e.g. the people working in this area will agree with your overview. You can cite sources beside published work (e.g., blogs, talks, etc).

There exists a commonly-held belief that convolutional neural networks require a large diversity of filters to perform well. This belief is reflected in how the networks are typically designed and trained: A large number of filters are individually trained in all SOTA models[27, 21, 24, 8, 15]. This number is often increased for better performance gains.

Filter diversity has been a focus of research for enhancing model performance and for network pruning and compression. For instance, Zhang et al. [31] proposed a new training strategy called "weight asynchronous update" to increase filter diversity in CNNs. Graff et al. [7] investigated the correlation between filter diversity and CNN accuracy, though their experiments were limited to first-layer filters. Gavrikov et al. [6] investigated trained filters of CNNs and found low diversity in structure of deep layer filters, noting them as redundant: "Filters are structurally similar to each other and therefore redundant."

This common belief in the necessity of filter diversity is often justified by the perceived need for a high variety of filters to capture all different features in datasets, an example of which can be seen in the response to this stackoverflow question[1].

B.2 How are your results in tension with this commonly-held position?

Detail how your submission challenges the belief described in (1). You may cite or synthesize results (e.g. figures, derivations, etc) from the main body of your submission and/or the literature.

Our paper challenges this widely held belief through a systematic search for a minimal high-performing filter set. Our findings reveal that only 8 filters are necessary for competitive performance, which stands in stark contrast to the thousands of filters typically used in state-of-the-art models.

We demonstrate that all filters of ConvNeXt-V2 models, regardless of their parameter count, can be linearly approximated by one of these 8 filters. This approximation maintains reasonable performance without any fine-tuning (Table 3). Furthermore, we show that training models with these 8 frozen filter types plus a bias can achieve very competitive results (Table 1). These findings challenge the necessity of training thousands of individual filters.

It's important to note that feature map diversity should be distinguished from filter diversity. Our research specifically questions the latter. We posit that a limited set of filters can produce various features when combined in subsequent layers or with weighted sums (pointwise layers) in the same layer.

B.3 How do you expect your submission to affect future work?

Perhaps the new understanding you are proposing calls for new experiments or theory in the area, or maybe it casts doubt on a line of research.

Our discovery of only 8 filters, all of which are mathematically describable, opens up a new line of research on mathematical theories for understanding and explaining how CNNs work. This simplification of filter diversity could provide a more accessible framework for theoretical analysis of CNN operations.

Moreover, with the drastic decrease in filter variety across all layers, we cast doubt on the commonly accepted transition from general to specialized filters in the deeper layers of CNNs. These results call for more experiments on filter generalization, potentially challenging long-held assumptions about the nature of feature extraction in deep networks.

Finally, we are optimistic that our findings will be beneficial for designing new CNN architectures that are more efficient or better performing. These potential impacts collectively suggest a paradigm shift in how we think about and design CNNs, calling for a reevaluation of current practices and opening up new avenues for both theoretical and applied research.