An Empirical Investigation of Model-to-Model Distribution Shifts in Trained Convolutional Filters

Anonymous Author(s)
Affiliation
Address
email

Abstract

We present first empirical results from our ongoing investigation of distribution shifts in image data used for various computer vision tasks. Instead of analyzing the original training and test data, we propose to study shifts in the learned weights of trained models. In this work, we focus on the properties of the distributions of dominantly used 3×3 convolution filter kernels. We collected and publicly provide a data set with over half a billion filters from hundreds of trained CNNs, using a wide range of data sets, architectures, and vision tasks. Our analysis shows interesting distribution shifts (or the lack thereof) between trained filters along different axes of meta-parameters, like data type, task, architecture, or layer depth. We argue, that the observed properties are a valuable source for further investigation into a better understanding of the impact of shifts in the input data to the generalization abilities of CNN models and novel methods for more robust transfer-learning in this domain.

1 Introduction

Despite their overwhelming success in the application to various vision tasks, the practical deployment of convolutional neural networks (CNNs) is still suffering from several inherent drawbacks. Two prominent examples are I) the dependence on very large amounts of annotated training data [1], which is not available for all target domains and is expensive to generate; and II) still widely unsolved problems with the robustness and generalization abilities of CNNs [2] towards shifts of the input data distributions. One can argue that both problems are strongly related, since a common practical solution to I) is the fine-tuning [3] of pre-trained models by small data sets from the actual target domain. This results in the challenge to find suitable pre-trained models based on data distributions that are "as close as possible" to the target distributions. Hence, both cases (I+II) imply the need to model and observe distribution shifts in the contexts of CNNs.

In this paper, we propose not to investigate these shifts in the input (image) domain, but rather in the weight distributions of the CNNs themselves. We argue that e.g. the distributions of trained convolutional filters in a CNN, which implicitly reflect the sub-distributions of the input image data which are actually utilized by a specific model, are more suitable and easier accessible representations for this task.

2 Methods

Data. We collected a total of **391 publicly available CNN models** pre-trained for various visual tasks, recorded meta-data for each model, and manually categorized the training data into visually distinctive groups like *natural scenes, medical ct, seismic,* or *astronomy* for example. All models were trained with full 32-bit precision but may have been trained with variously scaled inputs. The dominant subset is formed by image classification models trained on *ImageNet1k* [4] (264 models). We extracted all trained convolution filters to get a heterogeneous and diverse representation. Hereby, only the widely used filters with a kernel size of 3×3 were taken into account. Filters were only

Submitted to 35th Conference on Neural Information Processing Systems (NeurIPS 2021). Do not distribute.

extracted from regular convolution layers; Transposed convolution layers were not included. A total of **524,563,289 filters** from **13,015 layers** is used for the following study[1].

Structure analysis. We apply a full-rank PCA transformation to understand the underlying structure of the filters. A linear combination of principal components/eigenimages/basis vectors v_i weighted by the coefficients c_i: $f = \sum_i c_i v_i$ then describes each filter f. Figure 1 shows the principal components computed on the complete data set and various sub-sets.

Measuring distribution drifts. The divergence between two distributions is measured by the symmetric, non-negative variant of Kullback-Leibler [5]. The drift D of two filter sets is then defined by the sum of the divergence of the coefficient distributions P_i, Q_i along every principal component index i. The sum is weighted by the ratio of variance q_i explained by the i-th principal component.

$$D(P \parallel Q) = \sum_i q_i \sum_{x \in \mathcal{X}} P_i(x) \log \left(\frac{P_i(x)}{Q_i(x)} \right) + Q_i(x) \log \left(\frac{Q_i(x)}{P_i(x)} \right) \tag{1}$$

To avoid undefined expressions, all probability distributions F are set to hold $\forall x \in \mathcal{X} : F(x) \geq \epsilon$.

3 Empirical evaluation

3.1 Comparison of filter structures

In a first series of experiments, we analyze only the structure of 3×3 filters, neglecting their actual numerical weight in the trained models. We scale all filters by their absolute maximum weight and perform a PCA. Figure 1 shows some qualitative examples of principal components, split by several meta-data dimensions. Figure 2 depicts the shifts between distributions of PCA coefficients along models trained on different data types. We hypothesize that the spiky kernel density estimates (KDEs) are caused by "degenerated" layers e.g. if a model was too complex/deep for a given dataset (e.g. *ResNet-101* [6] on *CIFAR-10* [7]).

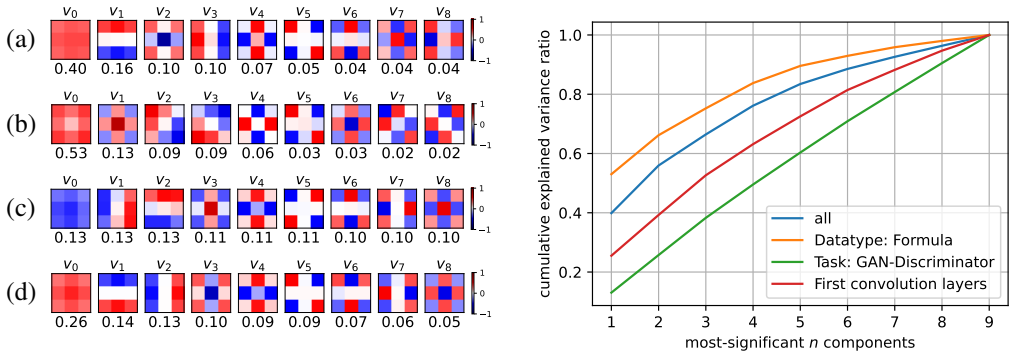


Figure 1: Selected eigenimages v_i and (cumulative) explained variance ratio per component for filters from (a) all, (b) models trained on *formula* data, (c) GAN discriminators, (d) first convolution layers. More examples in subsection A.3.

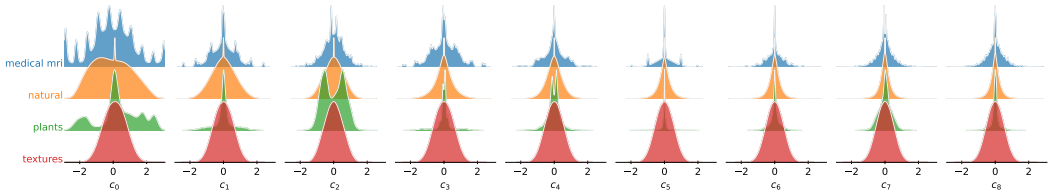


Figure 2: Coefficient distributions along the principal components for selected data types. Full overview in subsection A.5.

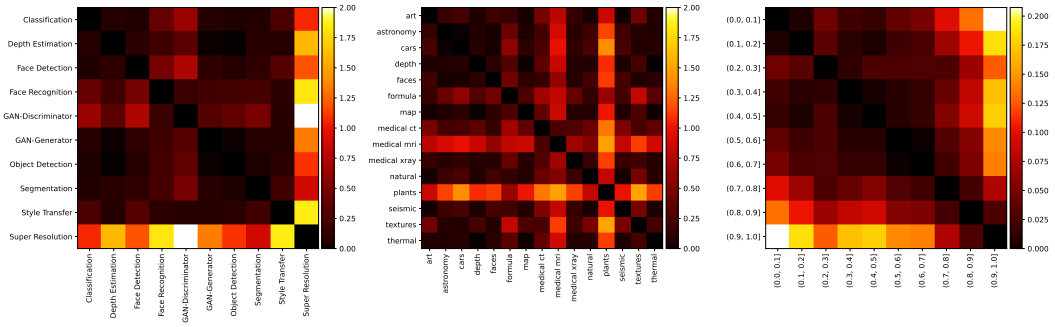
[1]Data set and code will be released publicly after acceptance, as an anonymous upload is not manageable.

3.1.1 Observable shifts along filters sub-sets

Between Models. It is not surprising, that the comparison of filters between individual neural architectures results in the highest shift we observe for all investigated meta-groups (Figure 4).

Inside Model families. The shift between models of the same family trained for the same task (e.g. *ResNet*-classifiers in Figure 4) is negligible, indicating that the common practice of pre-training models with ImageNet even for visually distant application domains is indeed a valid approach.

Between Tasks. Unsurprisingly, classification, segmentation, object detection, and GAN-generator distributions are quite similar (Figure 3a), since the non-classification models typically include a classification backbone. The smallest mean shift to other tasks is observed in object detection, GAN-generators, depth estimation models. Super Resolution models are strong outliers. We assume that this is due to the usage of *PixelShuffle* layers [8]. Less transferable task distributions also include GAN-discriminators and face detection. GAN-discriminators distributions do barely differ along principal components and can be approximated by a gaussian distribution. This indicates a filter distribution close to random initialization in many cases, resulting in a "confused" discriminator that cannot distinguish between real and fake samples towards the end of training.



(a) (b) (c)

Figure 3: D matrices for different filter meta-groups: (a) by tasks, (b) by data types, (c) by layer depth decile relative to the model depth.

Data types and training sets. We find that the distribution shift is well balanced across most data types and training sets (Figure 3b). Most coefficient distributions tend to shrink towards the least significant principal components, with a remarkably wide distribution of the first principal component. Notable outliers include *medical CT & MRI, formula, texture,* and *plants* data types. *Medical* types have visible spikes in the KDEs, indicating that many structurally similar filters exist. *Formula* includes models trained on *Fractal-DB*, which was proposed as a synthetic pre-training alternative to *ImageNet1k* [9].

Layer depth. The shift between layers of various depth deciles increases with the difference in depth (Figure 3c), yet it is marginal compared to the shift across tasks or data types. Distributions in the last decile of depth form the most distinct interval, significantly outdistancing the first and second-to-last decile that follow next. However, splitting the coefficients by absolute depth introduces many extreme outliers (Figure 4). That may again be a result of degenerated layers.

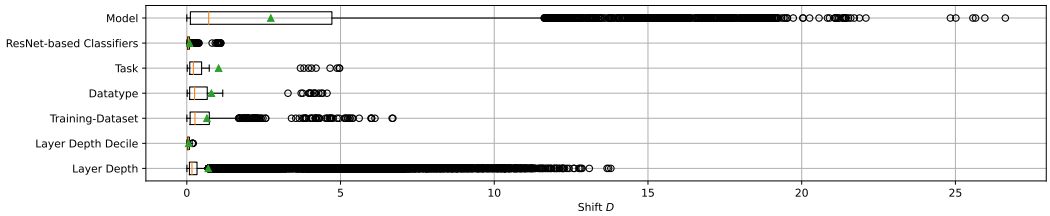
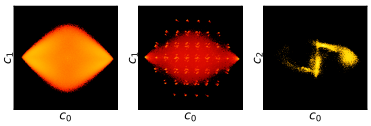


Figure 4: Distribution of the pair-wise shift D for different filter sub-sets.

3.1.2 Observable filter distribution phenotypes

We categorize the distributions into three phenotypes depending on their distribution characteristic in the PCA space: I) distributions where all dimensions are gaussian-like; II) distributions containing one or more layer that shows a fairly small variety of feature patterns and therefore takes on discrete stages in bi-variate scatter plots; III) distributions where one or more distribution is multi-modal, not centered, highly sparse or otherwise non-normal forming scatter plots that look like symbols. Figure 5 shows typical examples of these phenotypes.



(a) (b) (c)

Figure 5: Bi-variate plot between component distributions categorized as (a) sun, (b) spikes, (c) symbols.

3.2 Filter Scales

So far we have only studied the structural similarity, independent of the actual scale of the learned filter weights (difference between minimal and maximal weight). In Figure 6 we compute the mean scale per layer depth decile. The distributions show an expected decrease with depth but also a high variance and many outliers across models, especially in the first two deciles.

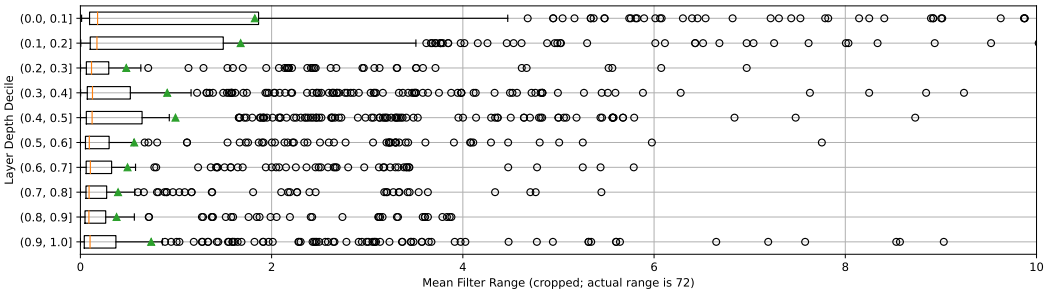


Figure 6: Boxplots showing mean range per layer depth decile (top to bottom in decreasing order) for each model in the dataset.

4 Related Work

An extensive analysis of features, connections, and their organization extracted from trained *InceptionV1* [10] was presented in [11, 12, 13, 14, 15, 16, 17, 18, 19]. The authors of [20] studied learned filter representations in ImageNet classification models and presented the first moves towards transfer learning. A summary of transfer learning for image classification CNN can be found in [21] and general surveys for other tasks and domains are available in [22, 23]. [24] captured convolution filter patterns with Gaussian Mixture Models to achieve cross-architecture transfer learning and [25] demonstrated that convolutions filters can be replaced by a fixed filter basis that 1×1 convolution layers blend.

A benchmark for distribution shifts that arise in real-world applications is provided in [26] and [27] measured robustness to natural distribution shifts. Lastly, [28] studied the correlation between transfer performance and distribution shifts of image classification models.

5 Discussion and Outlook

Our first results support our initial hypothesis that the distributions of trained convolutional filters are a suitable and easy-to-access proxy for the investigation of image distributions and the similarity between the same. While the presented results are still in the early stages of a thorough study, we report several interesting findings that could be explored to obtain better model generalizations and assist in finding suitable pre-trained models. One finding is the presence of large amounts of degenerated (or untrained) filters in large, well-performing networks - resulting in the phenotypes *spikes* and *symbols*. We assume that their existence is a symptom in line with the *Lottery Ticket Hypothesis* [29].

Another striking finding is the observation of very low shifts between different meta-groups: I) shifts inside a family of architectures are very low, independent of the target image distribution; II) also we observe rather small shifts between convolution layers of different depths; while III) shifts between different tasks are much higher, even when related architectures and the same data is used.

References

[1] Chen Sun, Abhinav Shrivastava, Saurabh Singh, and Abhinav Gupta. Revisiting unreasonable effectiveness of data in deep learning era. In *Proceedings of the IEEE international conference on computer vision*, pages 843–852, 2017.

[2] Naveed Akhtar and Ajmal Mian. Threat of adversarial attacks on deep learning in computer vision: A survey. *Ieee Access*, 6:14410–14430, 2018.

[3] Nima Tajbakhsh, Jae Y Shin, Suryakanth R Gurudu, R Todd Hurst, Christopher B Kendall, Michael B Gotway, and Jianming Liang. Convolutional neural networks for medical image analysis: Full training or fine tuning? *IEEE transactions on medical imaging*, 35(5):1299–1312, 2016.

[4] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE Conference on Computer Vision and Pattern Recognition*, pages 248–255, 2009.

[5] S. Kullback and R. A. Leibler. On Information and Sufficiency. *The Annals of Mathematical Statistics*, 22(1):79 – 86, 1951.

[6] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition, 2015.

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

[8] Wenzhe Shi, Jose Caballero, Ferenc Huszár, Johannes Totz, Andrew P. Aitken, Rob Bishop, Daniel Rueckert, and Zehan Wang. Real-time single image and video super-resolution using an efficient sub-pixel convolutional neural network, 2016.

[9] Hirokatsu Kataoka, Kazushige Okayasu, Asato Matsumoto, Eisuke Yamagata, Ryosuke Yamada, Nakamasa Inoue, Akio Nakamura, and Yutaka Satoh. Pre-training without natural images. 2020.

[10] Christian Szegedy, Wei Liu, Yangqing Jia, Pierre Sermanet, Scott Reed, Dragomir Anguelov, Dumitru Erhan, Vincent Vanhoucke, and Andrew Rabinovich. Going deeper with convolutions, 2014.

[11] Chris Olah, Nick Cammarata, Ludwig Schubert, Gabriel Goh, Michael Petrov, and Shan Carter. Zoom in: An introduction to circuits. *Distill*, 5, 2020.

[12] Chris Olah, Nick Cammarata, Ludwig Schubert, Gabriel Goh, Michael Petrov, and Shan Carter. An overview of early vision in inceptionv1. *Distill*, 2020. https://distill.pub/2020/circuits/early-vision.

[13] Nick Cammarata, Gabriel Goh, Shan Carter, Ludwig Schubert, Michael Petrov, and Chris Olah. Curve detectors. *Distill*, 2020. https://distill.pub/2020/circuits/curve-detectors.

[14] Chris Olah, Nick Cammarata, Chelsea Voss, Ludwig Schubert, and Gabriel Goh. Naturally occurring equivariance in neural networks. *Distill*, 2020. https://distill.pub/2020/circuits/equivariance.

[15] Ludwig Schubert, Chelsea Voss, Nick Cammarata, Gabriel Goh, and Chris Olah. High-low frequency detectors. *Distill*, 2021. https://distill.pub/2020/circuits/frequency-edges.

[16] Nick Cammarata, Gabriel Goh, Shan Carter, Chelsea Voss, Ludwig Schubert, and Chris Olah. Curve circuits. *Distill*, 2021. https://distill.pub/2020/circuits/curve-circuits.

[17] Chelsea Voss, Nick Cammarata, Gabriel Goh, Michael Petrov, Ludwig Schubert, Ben Egan, Swee Kiat Lim, and Chris Olah. Visualizing weights. *Distill*, 2021. https://distill.pub/2020/circuits/visualizing-weights.

[18] Chelsea Voss, Gabriel Goh, Nick Cammarata, Michael Petrov, Ludwig Schubert, and Chris Olah. Branch specialization. *Distill*, 2021. https://distill.pub/2020/circuits/branch-specialization.

[19] Michael Petrov, Chelsea Voss, Ludwig Schubert, Nick Cammarata, Gabriel Goh, and Chris Olah. Weight banding. *Distill*, 2021. https://distill.pub/2020/circuits/weight-banding.

[20] Jason Yosinski, Jeff Clune, Yoshua Bengio, and Hod Lipson. How transferable are features in deep neural networks? volume 4, 2014.

[21] Mahbub Hussain, Jordan J. Bird, and Diego R. Faria. A study on cnn transfer learning for image classification. In Ahmad Lotfi, Hamid Bouchachia, Alexander Gegov, Caroline Langensiepen, and Martin McGinnity, editors, *Advances in Computational Intelligence Systems*, pages 191–202, Cham, 2019. Springer International Publishing.

[22] Sinno Jialin Pan and Qiang Yang. A survey on transfer learning. *IEEE Transactions on Knowledge and Data Engineering*, 22(10):1345–1359, 2010.

[23] Fuzhen Zhuang, Zhiyuan Qi, Keyu Duan, Dongbo Xi, Yongchun Zhu, Hengshu Zhu, Hui Xiong, and Qing He. A comprehensive survey on transfer learning, 2020.

[24] Mehmet Aygun, Yusuf Aytar, and Hazim Kemal Ekenel. Exploiting convolution filter patterns for transfer learning. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV) Workshops*, Oct 2017.

[25] Muhammad Tayyab and Abhijit Mahalanobis. Basisconv: A method for compressed representation and learning in cnns, 2019.

[26] Pang Wei Koh, Shiori Sagawa, Henrik Marklund, Sang Michael Xie, Marvin Zhang, Akshay Balsubramani, Weihua Hu, Michihiro Yasunaga, Richard Lanas Phillips, Irena Gao, Tony Lee, Etienne David, Ian Stavness, Wei Guo, Berton Earnshaw, Imran Haque, Sara M Beery, Jure Leskovec, Anshul Kundaje, Emma Pierson, Sergey Levine, Chelsea Finn, and Percy Liang. Wilds: A benchmark of in-the-wild distribution shifts. 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 5637–5664. PMLR, 18–24 Jul 2021.

[27] Rohan Taori, Achal Dave, Vaishaal Shankar, Nicholas Carlini, Benjamin Recht, and Ludwig Schmidt. Measuring robustness to natural distribution shifts in image classification, 2020.

[28] Josip Djolonga, Jessica Yung, Michael Tschannen, Rob Romijnders, Lucas Beyer, Alexander Kolesnikov, Joan Puigcerver, Matthias Minderer, Alexander D'Amour, Dan Moldovan, Sylvain Gelly, Neil Houlsby, Xiaohua Zhai, and Mario Lucic. On robustness and transferability of convolutional neural networks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 16458–16468, June 2021.

[29] Jonathan Frankle and Michael Carbin. The lottery ticket hypothesis: Finding sparse, trainable neural networks. *arXiv preprint arXiv:1803.03635*, 2018.

[30] Herbert A. Sturges. The choice of a class interval. *Journal of the American Statistical Association*, 21, 1926.

[31] David Freedman and Persi Diaconis. On the histogram as a density estimator: L2 theory. *Zeitschrift für Wahrscheinlichkeitstheorie und Verwandte Gebiete*, 57, 1981.

A Appendix

A.1 Detailed divergence computation

All probability distributions are represented by histograms. The histogram range is defined by the minimum and maximum value found across all distributions to compare. The histogram widths are uniform and defined by the maximum bin width needed to represent each distribution. We use the maximum of the Sturges-[30] and Freedman-Diaconis-Estimator [31] as an estimate for the optimal bin width.

A.2 D-matrix for non-scaled filters

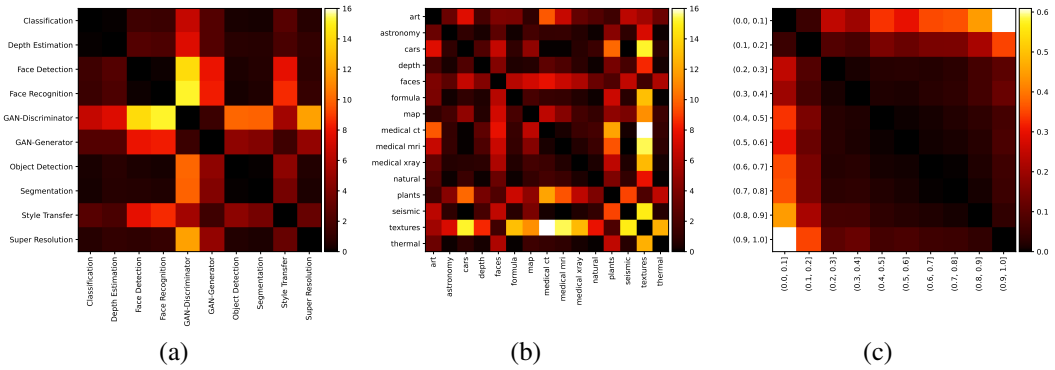


Figure 7: D matrices for different filter sub-sets on the raw filter data (non-scaled): (a) tasks, (b) data types, (c) filter depth decile relative to the model depth.

A.3 PCA

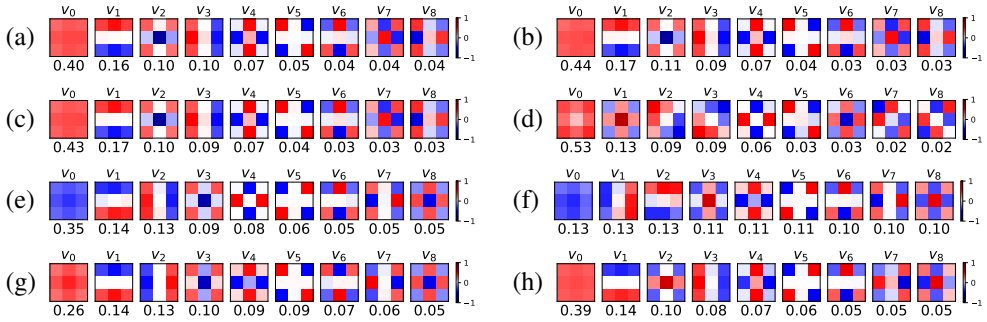
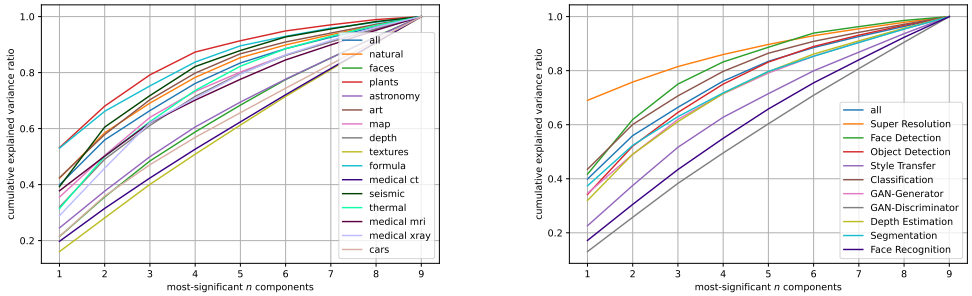


Figure 8: More eigenimages v_i and explained variance ratio per component for filters from (a) all, (b) classification models, (c) models trained on ImageNet1k, (d) models trained on formula data, (e) GAN generators, (f) GAN discriminators, (g) first convolution layers, (h) last convolution layers. Computed on the scaled data set.



(a) data type (b) task

Figure 9: Cumulative ration of explained variance over the first n components by all tasks and data types.

A.4 Data set statistics

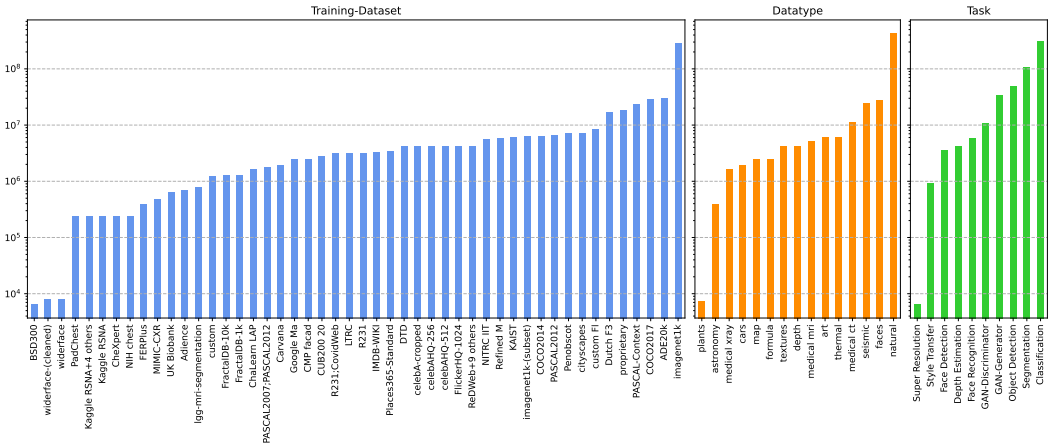


Figure 10: Total count of filters per filter sub-set. Log scale.

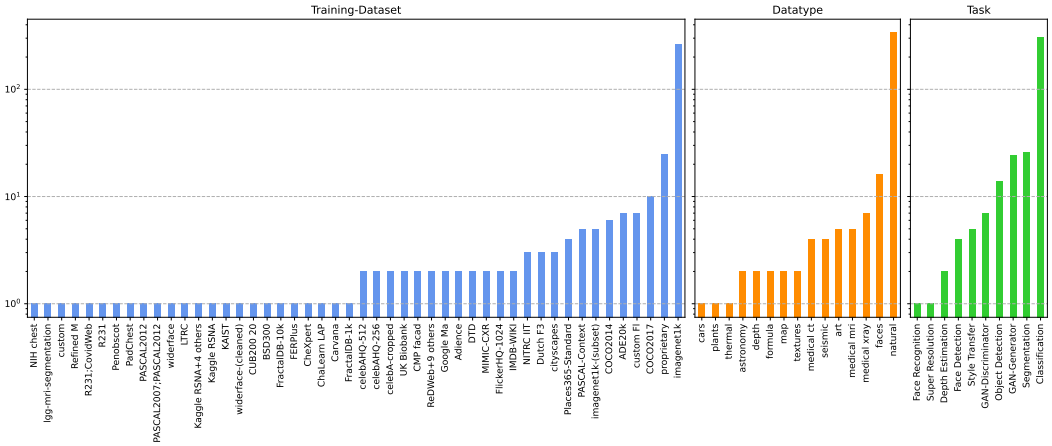


Figure 11: Total count of models per filter sub-set. Log scale.

A.5 Ridge Plots

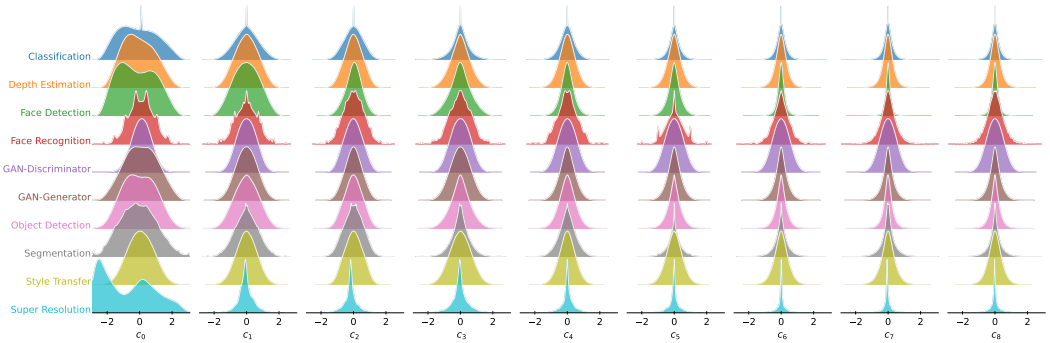


Figure 12: Distribution of the coefficients along the principal components by model task.

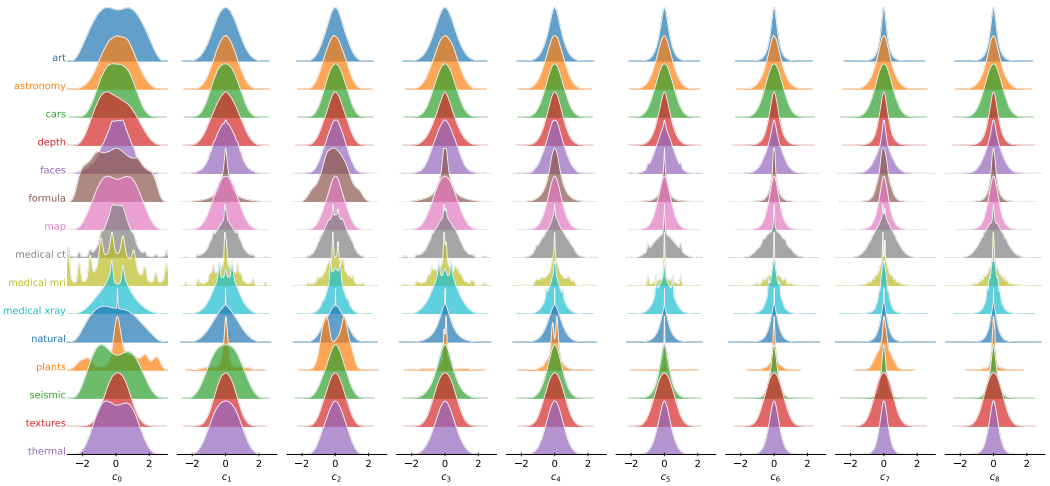


Figure 13: Distribution of the coefficients along the principal components by training data type.

Figure 14: Distribution of the coefficients along the principal components by combination of data set used for training.

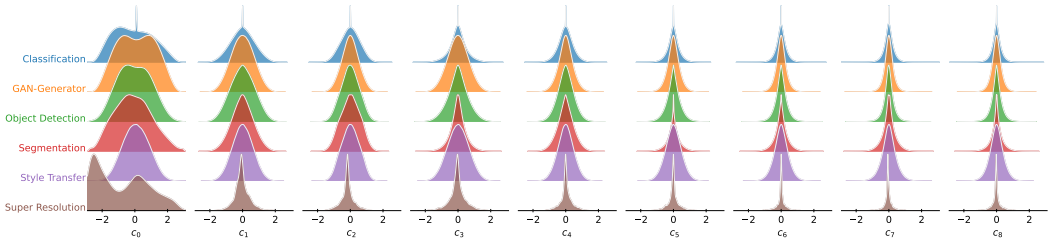


Figure 15: Distribution of the coefficients along the principal components by model task for data sets labeled as natural datatype.

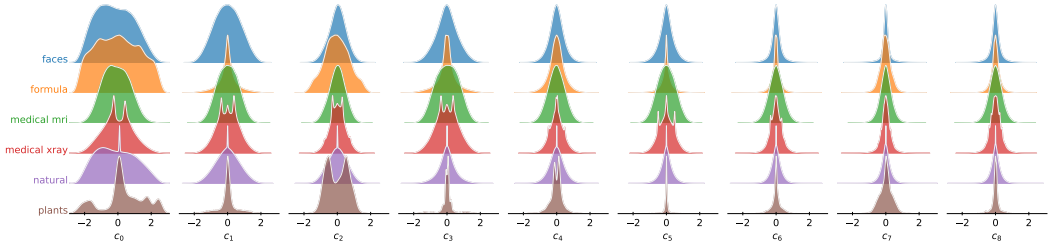


Figure 16: Distribution of the coefficients along the principal components by training datatype for image classification models.

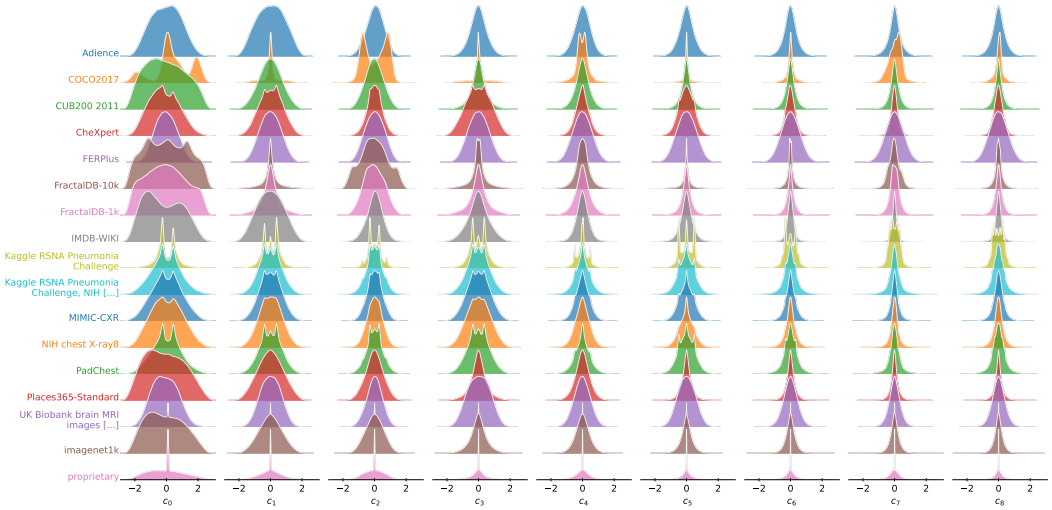


Figure 17: Distribution of the coefficients along the principal components by training data set for image classification models.

A.6 Model overview

A full overview over all used models will be made available on the data website after publication.