CNN Filter DB: An Empirical Investigation of Trained Convolutional Filters

Paul Gavrikov[1*] and Janis Keuper[1,2,3*]
[1]IMLA, Offenburg University, [2]CC-HPC, Fraunhofer ITWM, [3]Fraunhofer Research Center ML
{first.last}@hs-offenburg.de

Abstract

Currently, many theoretical as well as practically relevant questions towards the transferability and robustness of Convolutional Neural Networks (CNNs) remain unsolved. While ongoing research efforts are engaging these problems from various angles, in most computer vision related cases these approaches can be generalized to investigations of the effects of distribution shifts in image data.
In this context, we propose to study the shifts in the learned weights of trained CNN models. Here we focus on the properties of the distributions of dominantly used 3×3 convolution filter kernels. We collected and publicly provide a dataset with over 1.4 billion filters from hundreds of trained CNNs, using a wide range of datasets, architectures, and vision tasks. In a first use case of the proposed dataset, we can show highly relevant properties of many publicly available pre-trained models for practical applications: I) We analyze distribution shifts (or the lack thereof) between trained filters along different axes of meta-parameters, like visual category of the dataset, task, architecture, or layer depth. Based on these results, we conclude that model pre-training can succeed on arbitrary datasets if they meet size and variance conditions. II) We show that many pre-trained models contain degenerated filters which make them less robust and less suitable for fine-tuning on target applications.
Data & Project website: https://github.com/paulgavrikov/cnn-filter-db

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 datasets 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.

*Funded by the Ministry for Science, Research and Arts, Baden-Wuerttemberg, Grant 32-7545.20/45/1 (Q-AMeLiA).
The authors also thank Margret Keuper for her support and encouragement to submit this work.

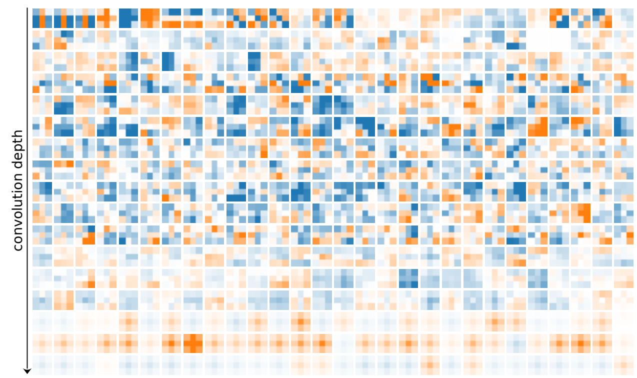


Figure 1. First 3×3 filters extracted of each convolution layer in a *ResNet-18* trained on *CIFAR-10*. The filters show a clear loss of diversity and increasing sparsity with depth. The colormap range is determined layer-wise by the absolute peak weight of all filters in that layer.

In this paper, we propose not to investigate these shifts in the input (image) domain, but rather in the 2D filter-kernel 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, are more suitable and easier accessible representations for this task. In order to foster systematic investigations of learned filters, we collected and publicly provide a dataset of over 1.4 billion filters with meta data from hundreds of trained CNNs, using a wide range of data sets, architectures, and vision tasks. To show the scientific value of this new data source, we conduct a first analysis and report a series of novel insights into widely used CNN models. Based on our presented methods we show that many publicly provided models suffer from degeneration. We show that overparameterization leads to sparse and/or non-diverse filters (Fig. 1), while robust training increases filter diversity, and reduces sparsity. Our results also show that learned filters do not significantly differ across models trained for various tasks, except for extreme outliers such as GAN-Discriminators. Models trained on datasets of different visual categories do not significantly drift either. Most shifts in studied models are due to degeneration, rather than an actual difference in structure. Therefore, our results imply that pre-training can be performed independent of the actual target data, and only the amount of training data and its diversity matters. This is inline with recent findings that models can be pre-trained even with images of fractals [4]. For classification models we show that the most variance in learned filters is found in the beginning and end of the model, while object/face detection models only show significant variance in early layers. Also, the most specialized filters are found in the last layers. We summarize our key contributions as follows:

- Publication of a diverse database of over 1.4B 3×3 convolution filters alongside with relevant meta information of the extracted filters and models [5].
- Presentation of a data-agnostic method based on sparsity and entropy of filters to find "degenerated" convolution layers due to overparameterization or non-convergence of trained CNN models.
- Showing that publicly available models often contain degenerated layers and can therefore be questionable candidates for transfer tasks.
- Analysis of distribution shifts in filters over various groups, providing insights that formed filters are fairly similar across a wide-range of examined groups.
- Showing that the model-to-model shifts that exist in classification models are, contrary to the predominant opinion, not only seen in deeper layers but also in the first layers.

Paper organization. We give an overview of our dataset and its collection process in Sec. 3, followed by an introduction of methods studying filter structure, distributions shifts, and layer degeneration such as randomness, low variance in filter structure, and high sparsity of filters. Then in Sec. 4 we apply these methods to our collected data. We show the impact of overparameterization and robust training on filter degeneration and provide intuitions for threshold finding. Then we analyze filter structures by determining a suitable filter basis and looking into reproducibility of filters in training, filter formation during training, and an analysis of distribution shifts for various dimensions of the collected meta-data. We discuss limitations of our approach in Sec. 5 and, finally, draw conclusions in Sec. 6.

2. Related Work

We are unaware of any systematic, large scale analysis of learned filters across a wide range of datasets, architectures and task such as the one performed in this paper. However, there are of course several partially overlapping aspects of our analysis that have been covered in related works:

Filter analysis. An extensive analysis of features, connections, and their organization extracted from trained *InceptionV1* [6] models was presented in [7–15]. The authors claim different CNNs will form similar features and circuits even when trained for different tasks.

Transfer learning. A survey on transfer learning for image classification CNNs can be found in [16] and general surveys for other tasks and domains are available in [17, 18]. The authors of [19] studied learned filter representations in *ImageNet1k* classification models and presented the first approaches towards transfer learning. They argued that different CNNs will form similar filters in early layers which will mostly resemble gabor-filters and color-blobs, while deeper layers will capture specifics of the dataset by forming increasingly specialized filters. [20] captured convolution filter pattern distributions with Gaussian Mixture Models to achieve cross-architecture transfer learning. [21] demonstrated that convolutions filters can be replaced by a fixed filter basis that 1×1 convolution layers blend.

Pruning criteria. Although we do not attempt pruning, our work overlaps with pruning techniques as they commonly rely on estimation criteria to understand which parameters to compress. These either rely on data-driven computation of a forward-pass [22–26], or backward-propagation [27, 28], or estimate importance solely based on the numerical weight (typically any ℓ-norm) of the parameters [29–33].

CNN distribution shifts. A benchmark for distribution shifts that arise in real-world applications is provided in [34] and [35] measured robustness to natural distribution shifts of 204 *ImageNet1k* models. The authors concluded that robustness to real-world shifts is low. Lastly, [36] studied the correlation between transfer performance and distribution shifts of image classification models and find that increasing training set and model capacity increases robustness to distribution shifts.

3. Methods

3.1. Collecting filters

We collected a total of **647 publicly available CNN models** from [37–39] and other sources that have been pre-trained for various 2D visual tasks[1]. In order to provide a heterogeneous and diverse representation of convolution filters "in the wild", we retrieved pre-trained models for 11 different tasks e.g. such as *classification*, *segmentation* and *image generation*. We also recorded various meta-data such as depth and frequency of included operations for each model, and manually categorized the variety of used training sets into 16 visually distinctive groups like *natural scenes, medical ct, seismic,* or *astronomy*. In total, the models were trained on 71 different datasets. The dominant subset is formed by *image classification* models trained on *ImageNet1k* [40] (355 models).

All models were trained with full 32-bit precision[2] but may have been trained on variously scaled input data. Included in the dataset are low-resolution variants of AlexNet [42], DenseNet-121/161/169 [43], ResNet-9/14/18/34/50/101/152 [44], VGG-11/13/16/19 [45], MobileNet v2 [46], Inception v3 [47] and GoogLeNet [6] image classification models that we have purposely trained on simple datasets such as CIFAR-10/100 [48], MNIST [49], Kuzushiji-MNIST (KMNIST) [50] and Fashion-MNIST [51] in order to study the effect of overparameterization on learned filters.

All collected models were converted into the ONNX format [52] which allows a streamlined filter extraction without framework dependencies. Hereby, only the widely used filters from regular convolution layers with a kernel size of 3×3 were taken into account. Transposed (sometimes also called de-convolution or up-convolution) convolution layers were not included. In total, **1,464,797,156 filters** from **21,436 layers** have been obtained for our dataset.

3.2. Analyzing filter structures

We apply a full-rank principal component analysis (PCA) transformation implemented via a singular-value decomposition (SVD) to understand the underlying structure of the filters [53].

First, we stack the relevant set of n flattened filters into a $n \times 9$ matrix X. Thereupon, we center the matrix and perform a SVD into a $n \times 9$ rotation matrix U, a 9×9 diagonal scaling matrix Σ, and a 9×9 rotation matrix V^T. The diagonal entries $\sigma_i, i = 0, \dots, n-1$ of Σ form the singular values in decreasing order of their magnitude. Row vectors $v_i, i = 0, \dots, n-1$ in V^T then form the principal components. Every row vector $c_{ij}, j = 0, \dots, n-1$ in U is the coefficient vector for f_i.

$$X^* = X - \bar{X} = U \Sigma V^T \tag{1}$$

Where $\bar{X}$ denotes the vector of column-wise mean values of any matrix X. Then we obtain a vector $\hat{a}$ of the explained variance ratio of each principal component. $\|\cdot\|_1$ denotes the ℓ_1-norm.

$$\begin{aligned} \vec{a} &= (\Sigma \cdot I)^2/(n-1) \\ \hat{a} &= \vec{a}/\|\vec{a}\|_1 \end{aligned} \tag{2}$$

Finally, each filter f' is described by a linear, shifted sum of principal components v_i weighted by the coefficients c_i.

$$f' = \sum_i c_i v_i + \bar{X}_i \tag{3}$$

3.3. Measuring distribution shifts

All probability distributions are represented by histograms. The histogram range is defined by the minimum and maximum value of all coefficients. Each histogram is divided into 70 uniform bins. The divergence between two distributions is measured by the symmetric, non-negative variant of Kullback-Leibler (KL_{sym}) [54].

$$\begin{aligned} KL(P \parallel Q) &= \sum_{x \in \mathcal{X}} P(x) \log \frac{P(x)}{Q(x)} \\ KL_{\text{sym}}(P \parallel Q) &= KL(P \parallel Q) + KL(Q \parallel P) \end{aligned} \tag{4}$$

We define the drift D between two filter sets 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 $\hat{a}_i$ explained by the i-th principal component.

$$D(P \parallel Q) = \sum_i \hat{a}_i \cdot KL_{\text{sym}}(P_i \parallel Q_i) \tag{5}$$

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

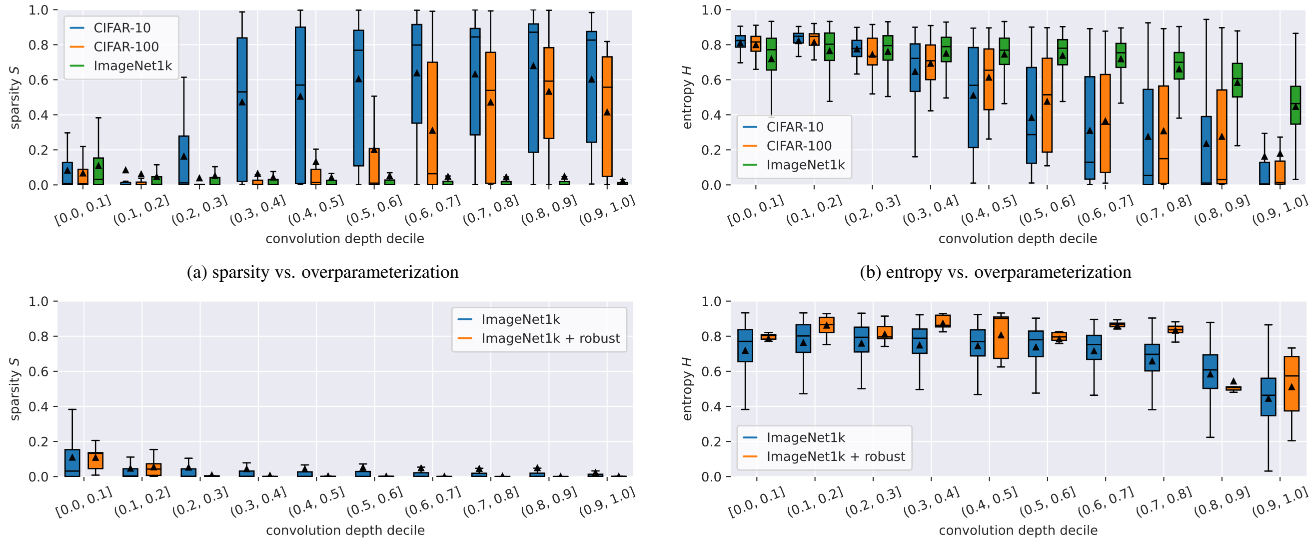
3.4. Measuring layer degeneration

Lottery Ticket Hypothesis [55] suggests that each architecture has a specific amount of convolution filters that saturate its ability to transform a given dataset into a well separable feature-space. Exceeding this number will result in a partitioning of the model into multiple inter-connected submodels. We hypothesize that these are seen in the form of degenerated filters in CNNs. In like manner, an insufficient amount of training samples or training epochs will also lead to degenerated filters. We characterize the following types of degeneration.

1. *High sparsity:* Filters are dominantly close to zero and therefore produce quasi-zero feature-maps [29]. These feature-maps carry no vital information and can be discarded.
2. *Low diversity in structure:* Filters are structurally similar to each other and therefore redundant. They produce similar feature-maps in different scales and could be represented by a subset of present filters.

[1] For more details refer to the supplementary materials.

[2] Although, initial experiments indicated that mixed/reduced precision training [41] does not affect distribution shifts beyond noise.



(a) sparsity vs. overparameterization

(b) entropy vs. overparameterization

(c) sparsity vs. robustness

(d) entropy vs. robustness

Figure 2. Comparison of layer entropy and sparsity of overparameterized, robust, and regular classification models. Outliers are hidden for clarity.

3. *Randomness:* Filter weights are conditionally independent of their neighbours. This indicates that no or not sufficient training was performed.

Sparsity degeneration is detectable by the share of sparse filters S in a given layer. We call a filter f sparse if all entries are near-zero. Consequently, given the number of input channels c_{in}, number of output channels c_{out}, and a set of filters in layer L, we can measure the layer sparsity by:

$$S(L) = \frac{|\{f | f \in L \wedge (\forall w \in f : -\epsilon_0 \leq w \leq \epsilon_0)\}|}{c_{\text{in}} c_{\text{out}}} \tag{6}$$

To detect the other types of degeneration we introduce a layer-wise metric based on the Shannon-Entropy of the explained variance ratio of each principal component obtained from a SVD of all filters in the examined layer (Sec. 3.2).

$$H = -\sum_i \hat{a}_i \log_{10} \hat{a}_i \tag{7}$$

If H is close to zero this indicates one strong principal component from which most of the filters can be reconstructed and is therefore a low filter diversity degeneration. On the other hand, a large entropy indicates a (close to) uniform distribution of the singular values and, thus, a randomness of the filters. Sparse layers are a specific form of low diversity degeneration and, generally both are correlated, whereas, sparsity and randomness are mutually exclusive. It should be noted, that $|\Sigma \cdot I| = \min(c_{\text{in}} c_{\text{out}}, 9)$ and therefore the entropy only becomes expressive if $c_{\text{in}} c_{\text{out}} \gg 9$.

4. Results: Analysis of trained CNN filters

4.1. Layer degeneration

In this section we study different causes of degeneration and aim provide thresholds for evaluation.

Overparameterization. The majority of the models that we have trained on our low resolution datasets are heavily overparameterized for these relatively simple problems. We base this argument on the fact that we have models with different depth for most architectures and already observe near perfect performance with the smallest variants. Therefore it is safe to assume that larger models are overparameterized especially given that the performance only increases marginally[1].
First we analyze layer sparsity and entropy for these models trained on *CIFAR-10/100* in comparison to all *ImageNet1k* classification models found in our dataset. For each dataset we have trained identical networks with identical hyperparameters. Both, *CIFAR-10* and *CIFAR-100*, consist of 60,000 32×32px images, but *CIFAR-100* includes 10x more labels and thus fewer samples per class forming a more challenging dataset.
Fig. 2a shows that the overparameterized models contain significantly more sparse filters on average, and that sparsity increases with depth. In particular, we see the most sparse filters for *CIFAR-10*. However, *ImageNet1k* classifiers also seem to have some kind of "natural" sparsity, even though we do not consider most of these models as overparameterized. Entropy, on the other hand, decreases with increasing layer depth for every *classifier*, but more rapidly in overpa-

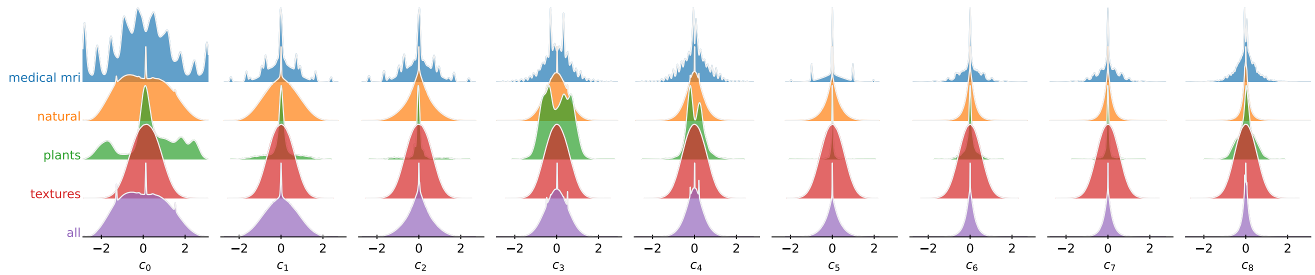


Figure 3. KDEs of the coefficient distributions along every principal component for selected[1] visual categories.

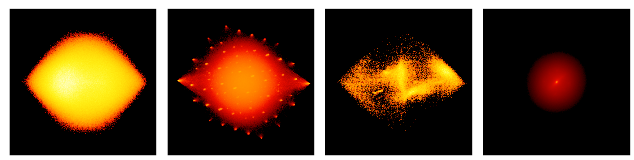
rameterized models (Fig. 2b). Again, the *CIFAR-10* models degrade faster and show more degeneration.
The overparameterized models contain layers that have a entropy close to 0 towards deeper layers which indicates that these models are "saturated" and only produce differently scaled variants of the same filters. In line with the oversaturation, these models also have increasingly sparse filters, presumably as an effect of regularization.

Filter degeneration and model robustness. Our dataset also contains robust models from the *RobustBench leaderboard* [38]. When comparing robust models with non-robust models trained on *ImageNet1k*, it becomes clear that robust models form almost no sparse filters after in deeper convolution layers (Fig. 2c), while regular models show some sparsity there. The entropy of robust models is also higher throughout depth (Fig. 2d), indicating that robust models learn more diverse filters.

Thresholds. To obtain a threshold for randomness given a number of filters n per layer we perform multiple experiments in which we initialize convolution filters of different sizes from a standard normal distribution and fit a sigmoid T_H to the minimum results obtained for entropy.

$$T_H(n) = \frac{L}{1 + e^{-k(\log_2(n) - x_0)}} + b \tag{8}$$

We obtain the following values $L = 1.26$, $x_0 = 2.30$, $k = 0.89$, $b = -0.31$ and call any layer L with $H > T_H(n)$ random. On the opposite, defining a threshold for low diversity degeneration seems less intuitive and one can only rely on statistics: The average entropy H is 0.69 over all layers and continuously decreases from an average of 0.75 to 0.5 with depth. Additionally, the minimum of the 1.5 IQR also steadily decreases with depth.
The same applies to sparsity: the average sparsity S over all layers is 0.12 and only 56.5% of the layers in our dataset hold $S < 0.01$ and 9.9% even show $S > 0.5$. In terms of convolution depth, the average sparsity varies between 9.9% and 14% with the largest sparsity found in the last 20% of the model depth. The largest outliers of the 1.5 interquartile range (IQR) are, however, found in the first decile. In both cases we find it difficult to provide a meaningful general threshold and suggest to determine this value on a case-by-case basis[1].



(a) sun (b) spikes (c) symbols (d) point

Figure 4. Bi-variate plots between component distributions showing the four phenotypes.

4.2. Filter structure

In the next series of experiments, we analyze only the structure of 3×3 filters, neglecting their actual numerical weight in the trained models. Therefore, we normalize each filter f individually by the absolute maximum weight into f'.

$$\begin{aligned} d_i &= \max_{i,j} |f_{ij}| \\ f'_{ij} &= \begin{cases} f_{ij}/d_i, & \text{if } d_i \neq 0 \\ f_{ij}, & \text{else} \end{cases} \end{aligned} \tag{9}$$

Then we perform a PCA transformation on the scaled filters. Fig. 5 shows some qualitative examples of obtained principal components, split by several meta-data dimensions. The images of the formed basis are often similar for all groups except for few outliers (such as *GAN-discriminators*). The explained variance however fluctuates significantly and sometimes changes the order of components. Consistently, we observe substantially higher variance on the first principal components. The explained variance does not necessarily correlate with the shift observed between models. Here, the biggest mean drift is also located in the first principal component ($\hat{D} = 0.90$), but is then followed by the sixth, third, second component ($\hat{D} = 0.78, 0.69, 0.58$). The coefficients of the sixth component also contain the strongest outliers (Fig. 6). We visualize the

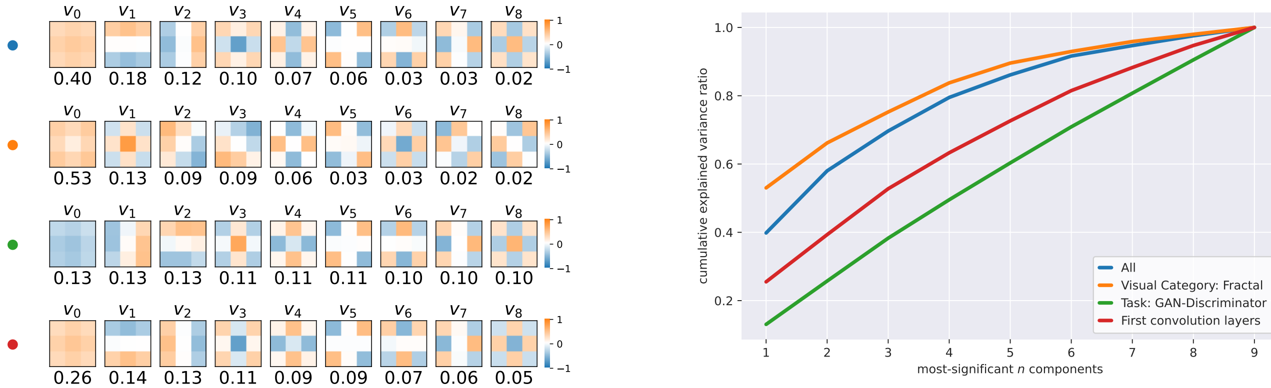


Figure 5. Selected[1] depiction of the filter basis and (cumulative) explained variance ratio per component for filters from • full dataset, • models trained on images of *fractals*, • GAN discriminators, • first convolution layers.

distributions of PCA coefficients along every component for each group by plots of kernel density estimates (KDEs), e.g. Fig. 3 depicts the distributions of filters grouped by some selected visual categories in comparison to the distribution of coefficients for the full dataset. Filters extracted from models with degenerated layers (as seen in *medical mri*) result in spiky/multi-modal KDEs. The distributions can alternatively be visualized by bi-variate scatter plots that may reveal more details than KDEs. For example, they let us categorize the distributions into phenotypes depending on their distribution characteristic in the PCA space (Fig. 4): *sun:* distributions where both dimensions are gaussian-like. These are to be expected coefficient distributions without significant sparsity/low diversity degeneration. Yet, this phenotype may also include non-converged filters; *spikes:* distributions suffering from a low variance degeneration resulting in local hotspots; *symbols:* at least one distribution is multi-modal, non-centered, highly sparse or otherwise non-normal (low variance degeneration); *point:* coefficients are primarily located in the center (sparsity degeneration).

Reproducibility of filters. We train low-resolution networks on *CIFAR-10* multiple times with identical hyper-parameters except for random seeds and save a checkpoints of each model at the best validation epoch. Most models are converging to highly similar coefficient distributions when retrained with different weight initialization (e.g. ResNet-9 with $D < 5.3 \cdot 10^{-4}$). However, some architectures such as *MobileNetv2* show higher shifts ($D < 2.6 \cdot 10^{-2}$). We assume that this is due to the structure of the loss surface, e.g. the residual skip connections found in *ResNets* smooth the surface, whereas other networks way contain more local minima due to noisy surfaces [56].

Formation of filter structures during training. Although our dataset only includes trained convolutional filters we tried to understand how the coefficient distribution shifts

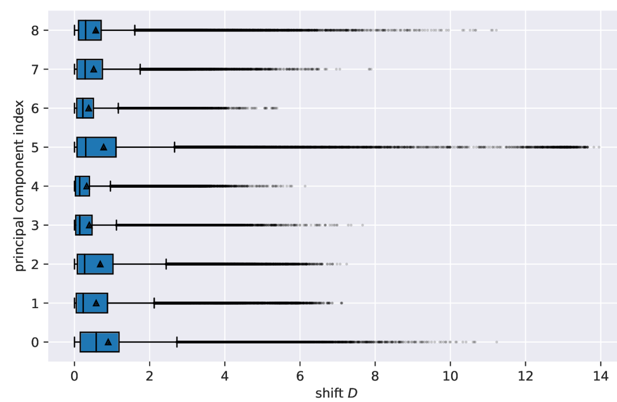


Figure 6. Distribution of the shift D along principal components computed on all possible pairings of models.

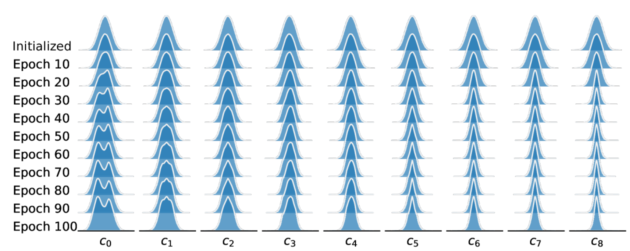
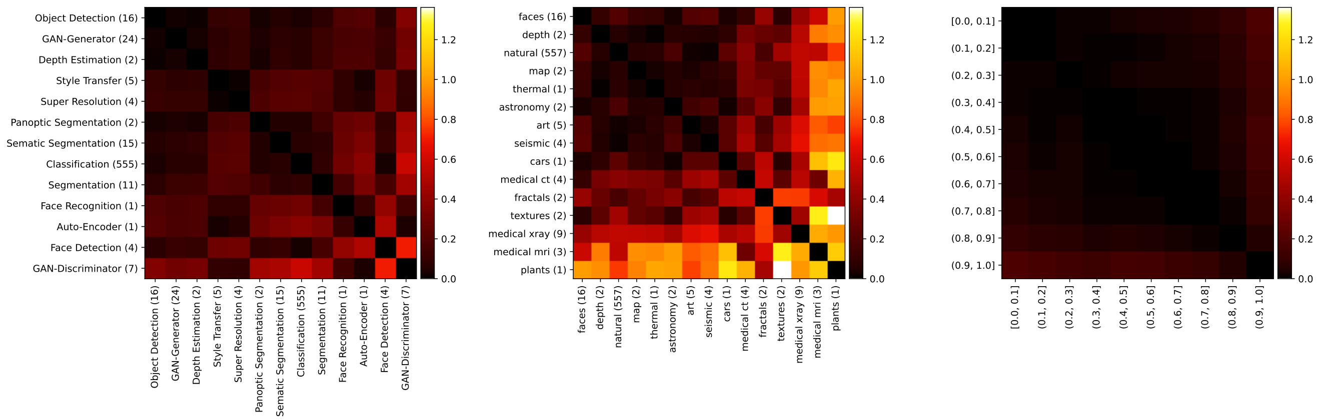


Figure 7. Coefficient distribution of a *ResNet-9* trained on *CIFAR-10* every 10 epochs.

during training. Therefore we recorded checkpoints of a *ResNet-9* trained on *CIFAR-10* every 10 training epochs beginning right after the weight initialization. Fig. 7 shows that the coefficient distributions along all principal components are gaussian-like distributed in the beginning and eventually shift during training. For this specific model, distributions along major principal components retain the standard deviation during training, while less-significant com-



(a) tasks (b) visual categories (c) convolution depth decile

Figure 8. Heatmaps over the shift D for different filters groupings. The number in brackets denotes the number of models in this group. Low values/dark colors denote low shifts.

ponent distributions decrease. The initialization observation helped us removing models from our collection where we failed to load the trained parameters for any reason and is foundation for our provided randomness metric.

4.3. Distribution shifts between trained models

In this subsection we are investigating transfer distance in different meta-dimensions of pre-trained models. We compute the shift D and visualize this is the form of heatmaps (Fig. 8) that show shifts between all pairings.

Shifts between tasks. Unsurprisingly, *classification*, *segmentation*, *object detection*, and *GAN-generator* distributions are quite similar, since the non-*classification* models typically include a *classification* backbone. The smallest mean shift to other tasks is observed in *object detection*, *GAN-generators*, and *depth estimation* models. The least transferable distributions are *GAN-discriminators*. Their distributions do barely differ along principal components and can be approximated by a gaussian distribution. By our randomness metric this indicates a filter distribution that is close to random initialization, implying a "confused" *discriminator* that cannot distinguish between real and fake samples towards the end of (successful) training. It may be surprising to see a slightly larger average shift for *classification*. This is presumably due to many degenerated layers in our collected models, which are also visible in the form of spikes when studying the KDEs. An evaluation[1] of distributions including only non-degenerated *classifiers* actually shows a lower average shift due to the aforementioned similarity to other tasks.

Shifts between visual categories and training sets. We find that the distribution shift is well balanced across most visual categories and training sets. Notable outliers include all *medical* types. They have visible spikes in the KDEs, once again indicating degenerated layers. Indeed, the average sparsity in these models is extreme in the last 80% of the model depth. Another interesting, albeit less significant outlier is the *fractal* category. It consists of models trained on *Fractal-DB*, which was proposed as a synthetic pre-training alternative to *ImageNet1k* [4]. The standard deviations of coefficient distributions tend to shrink towards the least significant principal components but this trend is not visible for this category indicating that sorting the basis by variance would yield a different order for this task and perhaps the basis itself is not well suited. Also notable is a remarkably high standard deviation on the distribution of the first principal component. Interestingly, we also observe sub-average degeneration for this category. Shifts in other categories can usually be explained by a biased representation. For example we only have one model for *plants*, our *handwriting* models consist exclusively of overparameterized networks that suffer from layer degeneration, and *textures* consists of only one *GAN-discriminator* which will naturally shows a high randomness.

Shifts by filter/layer depth. The shift between layers of various depth deciles increases with the difference in depth, with distributions in the last decile of depth forming the most distinct interval, and outdistancing the second-to-last and first decile that follow next. An interesting aspect is also the model-to-model shift across deciles. This shift exemplifies the uniqueness of formed filters. Our observations overhaul the general recommendation for fine-tuning to freeze early layers in *classification* models, as the largest shifts are not only seen in deep layers but also in early vision (Fig. 9). *Segmentation*[1] models show the most drift in deeper layers. Contrary, *object/face detection* models only

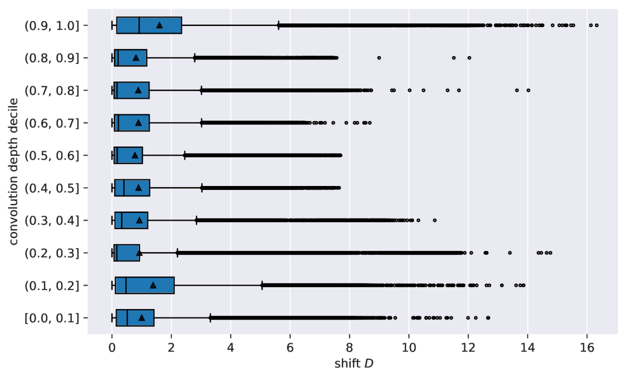


Figure 9. Boxplots showing the distribution of pair-wise model-to-model shift D of *classification* models per convolution depth decile (top to bottom in decreasing order). Our intentionally overparameterized models were left out of this analysis.

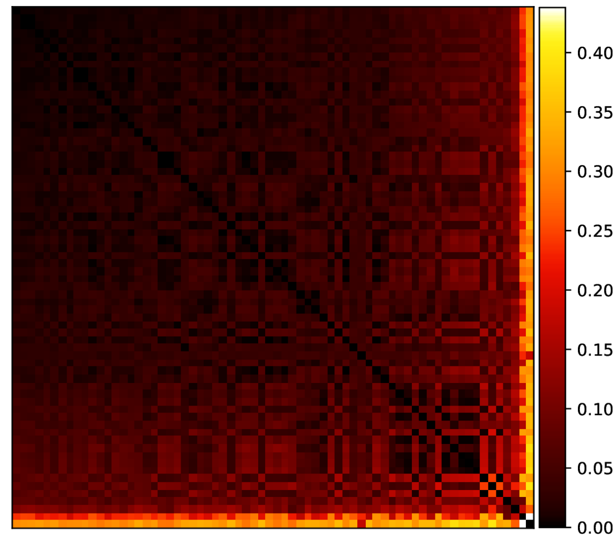


Figure 10. Heatmap over the shift D between different pairings of *ResNet-classifiers*. Each row/column depicts one model. Intentionally overparameterized models were not included.

show drift in the early vision (*object detection* in the first, *face detection* in the first four depth deciles), but marginal drift in later convolution stages.

Shifts within model families. The shift between models of the same family trained for the same task is negligible (Fig. 10), indicating that every large enough dataset is good enough and the common practice of pre-training models with *ImageNet1k* even for visually distant application domains is indeed a valid approach. *ResNet*-family outliers only consist out of models that show a high amount of sparsity. Additionally, this observation may be exploited by training small teacher networks and apply knowledge distillation [57] to initialize deeper models of the same family.

5. Limitations

Our data is biased against *classification* models and/or *natural* datasets such as *ImageNet1k*. Further, some splits will over-represent specific dimensions e.g. tasks may include exclusive visual categories and vice versa. Also, as previously shown, many of the collected models show a large amount of degenerated layers that impact the distributions. This also biases measurements of the distribution shifts. We performed an ablation study by removing filters extracted from degenerated layers, but were unable to find a clear correlation between degeneration and distribution shifts[1], presumably due to a lack of justified thresholds.

6. Conclusions

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 in the context of transferring pre-trained models and robustness. 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 for fine-tuning. One finding is the presence of large amounts of degenerated (or untrained) filters in large, well-performing networks - resulting in the phenotypes *points*, *spikes*, and *symbols*. We assume that their existence is a symptom in line with the *Lottery Ticket Hypothesis* [55]. We conclude that ideal models should have relatively high entropy (but $H < T_H$) throughout all layers and almost no sparse filters. Models that show an increasing or generally high sparsity or a massive surge in entropy with depth are most likely overparameterized and could be pruned, which would benefit inference and training speed. Whereas, initialized but not trained models will have a constantly high entropy $H \geq T_H$ throughout all layers and virtually no sparsity.

Another striking finding is the observation of very low shifts in filter structure between different meta-groups: I) shifts inside a family of architectures are very low; II) shifts are mostly independent of the target image distribution and task; III) also we observe rather small shifts between convolution layers of different depths with the highest shifts in the first and last layers. Overall, the analysis of over 1.4 billion learned convolutional filters in the provided dataset gives a strong indication that the common practice of pre-training CNNs is indeed a sufficient approach if the chosen model is not heavily overparameterized. Our first results indicate that the presented dataset is a rich source for further research in transfer learning, robustness and pruning.

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*, pp. 843–852, 2017. 1

[2] Naveed Akhtar and Ajmal Mian, "Threat of adversarial attacks on deep learning in computer vision: A survey," *Ieee Access*, vol. 6, pp. 14410–14430, 2018. 1

[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*, vol. 35, no. 5, pp. 1299–1312, 2016. 1

[4] Hirokatsu Kataoka, Kazushige Okayasu, Asato Matsumoto, Eisuke Yamagata, Ryosuke Yamada, Nakamasa Inoue, Akio Nakamura, and Yutaka Satoh, "Pre-training without natural images," 2020. 2, 7

[5] Paul Gavrikov and Janis Keuper, "CNN-Filter-DB v1.0.0," June 2022. Distributed by Zenodo. doi: https://doi.org/10.5281/zenodo.6371680. 2

[6] Christian Szegedy, Wei Liu, Yangqing Jia, Pierre Sermanet, Scott Reed, Dragomir Anguelov, Dumitru Erhan, Vincent Vanhoucke, and Andrew Rabinovich, "Going deeper with convolutions," 2014. 2, 3

[7] Chris Olah, Nick Cammarata, Ludwig Schubert, Gabriel Goh, Michael Petrov, and Shan Carter, "Zoom in: An introduction to circuits," *Distill*, vol. 5, 2020. 2

[8] 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. 2

[9] Nick Cammarata, Gabriel Goh, Shan Carter, Ludwig Schubert, Michael Petrov, and Chris Olah, "Curve detectors," *Distill*, 2020. https://distill.pub/2020/circuits/curve-detectors. 2

[10] 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. 2

[11] Ludwig Schubert, Chelsea Voss, Nick Cammarata, Gabriel Goh, and Chris Olah, "High-low frequency detectors," *Distill*, 2021. https://distill.pub/2020/circuits/frequency-edges. 2

[12] Nick Cammarata, Gabriel Goh, Shan Carter, Chelsea Voss, Ludwig Schubert, and Chris Olah, "Curve circuits," *Distill*, 2021. https://distill.pub/2020/circuits/curve-circuits. 2

[13] 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. 2

[14] Chelsea Voss, Gabriel Goh, Nick Cammarata, Michael Petrov, Ludwig Schubert, and Chris Olah, "Branch specialization," *Distill*, 2021. https://distill.pub/2020/circuits/branch-specialization. 2

[15] Michael Petrov, Chelsea Voss, Ludwig Schubert, Nick Cammarata, Gabriel Goh, and Chris Olah, "Weight banding," *Distill*, 2021. https://distill.pub/2020/circuits/weight-banding. 2

[16] Mahbub Hussain, Jordan J. Bird, and Diego R. Faria, "A study on cnn transfer learning for image classification," in *Advances in Computational Intelligence Systems* (Ahmad Lotfi, Hamid Bouchachia, Alexander Gegov, Caroline Langensiepen, and Martin McGinnity, eds.), (Cham), pp. 191–202, Springer International Publishing, 2019. 2

[17] Sinno Jialin Pan and Qiang Yang, "A survey on transfer learning," *IEEE Transactions on Knowledge and Data Engineering*, vol. 22, no. 10, pp. 1345–1359, 2010. 2

[18] Fuzhen Zhuang, Zhiyuan Qi, Keyu Duan, Dongbo Xi, Yongchun Zhu, Hengshu Zhu, Hui Xiong, and Qing He, "A comprehensive survey on transfer learning," 2020. 2

[19] Jason Yosinski, Jeff Clune, Yoshua Bengio, and Hod Lipson, "How transferable are features in deep neural networks?," vol. 4, 2014. 2

[20] 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. 2

[21] Muhammad Tayyab and Abhijit Mahalanobis, "Basisconv: A method for compressed representation and learning in cnns," 2019. 2

[22] Guillaume Alain and Yoshua Bengio, "Understanding intermediate layers using linear classifier probes," 2018. 2

[23] Jian-Hao Luo, Jianxin Wu, and Weiyao Lin, "Thinet: A filter level pruning method for deep neural network compression," 2017. 2

[24] Yihui He, Xiangyu Zhang, and Jian Sun, "Channel pruning for accelerating very deep neural networks," in *2017 IEEE International Conference on Computer Vision (ICCV)*, pp. 1398–1406, 2017. 2

[25] Shaohui Lin, Rongrong Ji, Yuchao Li, Yongjian Wu, Feiyue Huang, and Baochang Zhang, "Accelerating convolutional networks via global & dynamic filter pruning," in *Proceedings of the Twenty-Seventh International Joint Conference on Artificial Intelligence, IJCAI-18*, pp. 2425–2432, International Joint Conferences on Artificial Intelligence Organization, 7 2018. 2

[26] Shaohui Lin, Rongrong Ji, Chenqian Yan, Baochang Zhang, Liujuan Cao, Qixiang Ye, Feiyue Huang, and David Doermann, "Towards optimal structured cnn pruning via generative adversarial learning," in *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 2785–2794, 2019. 2

[27] Ruichi Yu, Ang Li, Chun-Fu Chen, Jui-Hsin Lai, Vlad I. Morariu, Xintong Han, Mingfei Gao, Ching-Yung Lin, and Larry S. Davis, "Nisp: Pruning networks using neuron importance score propagation," in *2018 IEEE/CVF Conference*

on Computer Vision and Pattern Recognition, pp. 9194–9203, 2018. 2

[28] Xiaohan Ding, Guiguang Ding, Yuchen Guo, and Jungong Han, "Centripetal sgd for pruning very deep convolutional networks with complicated structure," in *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 4938–4948, 2019. 2

[29] 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 *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, (Los Alamitos, CA, USA), pp. 2795–2804, IEEE Computer Society, jun 2019. 2, 3

[30] Song Han, Jeff Pool, John Tran, and William J. Dally, "Learning both weights and connections for efficient neural networks," 2015. 2

[31] Hao Li, Asim Kadav, Igor Durdanovic, Hanan Samet, and Hans Peter Graf, "Pruning filters for efficient convnets," 2017. 2

[32] Yang He, Ping Liu, Ziwei Wang, Zhilan Hu, and Yi Yang, "Filter pruning via geometric median for deep convolutional neural networks acceleration," in *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 4335–4344, 2019. 2

[33] Yang He, Guoliang Kang, Xuanyi Dong, Yanwei Fu, and Yi Yang, "Soft filter pruning for accelerating deep convolutional neural networks," in *Proceedings of the Twenty-Seventh International Joint Conference on Artificial Intelligence, IJCAI-18*, pp. 2234–2240, International Joint Conferences on Artificial Intelligence Organization, 7 2018. 2

[34] 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 *Proceedings of the 38th International Conference on Machine Learning* (Marina Meila and Tong Zhang, eds.), vol. 139 of *Proceedings of Machine Learning Research*, pp. 5637–5664, PMLR, 18–24 Jul 2021. 2

[35] Rohan Taori, Achal Dave, Vaishaal Shankar, Nicholas Carlini, Benjamin Recht, and Ludwig Schmidt, "Measuring robustness to natural distribution shifts in image classification," 2020. 2

[36] 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)*, pp. 16458–16468, June 2021. 2

[37] Ross Wightman, "Pytorch image models." https://github.com/rwightman/pytorch-image-models, 2019. 2

[38] Francesco Croce, Maksym Andriushchenko, Vikash Sehwag, Edoardo Debenedetti, Nicolas Flammarion, Mung Chiang, Prateek Mittal, and Matthias Hein, "Robustbench: a standardized adversarial robustness benchmark," in *Thirty-fifth Conference on Neural Information Processing Systems Datasets and Benchmarks Track*, 2021. 2, 5

[39] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Kopf, Edward Yang, Zachary DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala, "Pytorch: An imperative style, high-performance deep learning library," in *Advances in Neural Information Processing Systems 32* (H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett, eds.), pp. 8024–8035, Curran Associates, Inc., 2019. 2

[40] 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*, pp. 248–255, 2009. 3

[41] Paulius Micikevicius, Sharan Narang, Jonah Alben, Gregory Diamos, Erich Elsen, David Garcia, Boris Ginsburg, Michael Houston, Oleksii Kuchaiev, Ganesh Venkatesh, and Hao Wu, "Mixed precision training," 2018. 3

[42] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton, "Imagenet classification with deep convolutional neural networks," in *Advances in Neural Information Processing Systems* (F. Pereira, C. J. C. Burges, L. Bottou, and K. Q. Weinberger, eds.), vol. 25, Curran Associates, Inc., 2012. 3

[43] Gao Huang, Zhuang Liu, Laurens van der Maaten, and Kilian Q. Weinberger, "Densely connected convolutional networks," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, July 2017. 3

[44] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun, "Deep residual learning for image recognition," 2015. 3

[45] Karen Simonyan and Andrew Zisserman, "Very deep convolutional networks for large-scale image recognition," 2015. 3

[46] Mark Sandler, Andrew Howard, Menglong Zhu, Andrey Zhmoginov, and Liang-Chieh Chen, "Mobilenetv2: Inverted residuals and linear bottlenecks," 2019. 3

[47] Christian Szegedy, Vincent Vanhoucke, Sergey Ioffe, Jon Shlens, and Zbigniew Wojna, "Rethinking the inception architecture for computer vision," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2016. 3

[48] Alex Krizhevsky, Vinod Nair, and Geoffrey Hinton, "Cifar-10 (canadian institute for advanced research)," 3

[49] Li Deng, "The mnist database of handwritten digit images for machine learning research [best of the web]," *IEEE Signal Processing Magazine*, vol. 29, no. 6, pp. 141–142, 2012. 3

[50] Tarin Clanuwat, Mikel Bober-Irizar, Asanobu Kitamoto, Alex Lamb, Kazuaki Yamamoto, and David Ha, "Deep learning for classical japanese literature," 2018. 3

[51] Han Xiao, Kashif Rasul, and Roland Vollgraf, "Fashion-mnist: a novel image dataset for benchmarking machine learning algorithms," 2017. 3

[52] Junjie Bai, Fang Lu, Ke Zhang, *et al.*, "Onnx: Open neural network exchange." https://github.com/onnx/onnx, 2019. 3

[53] I Jolliffe, *Principal Component Analysis*. New York, NY: Springer New York, 1986. 3

[54] S. Kullback and R. A. Leibler, "On Information and Sufficiency," *The Annals of Mathematical Statistics*, vol. 22, no. 1, pp. 79 – 86, 1951. 3

[55] Jonathan Frankle and Michael Carbin, "The lottery ticket hypothesis: Finding sparse, trainable neural networks," *arXiv preprint arXiv:1803.03635*, 2018. 3, 8

[56] Lifu Wang, Bo Shen, Ning Zhao, and Zhiyuan Zhang, "Is the skip connection provable to reform the neural network loss landscape?," 2020. 6

[57] Geoffrey Hinton, Oriol Vinyals, and Jeff Dean, "Distilling the knowledge in a neural network," 2015. 8