Investigating Calibration and Corruption Robustness of Post-hoc Pruned Perception CNNs: An Image Classification Benchmark Study

Pallavi Mitra
Continental AG, Germany
first.last@continental.com

Gesina Schwalbe
University of Lübeck, Germany
first.last@uni-luebeck.de

Nadja Klein
TU Dortmund, Germany
first.last@tu-dortmund.de

Abstract

Convolutional Neural Networks (CNNs) have achieved state-of-the-art performance in many computer vision tasks. However, high computational and storage demands hinder their deployment into resource-constrained environments, such as embedded devices. Model pruning helps to meet these restrictions by reducing the model size, while maintaining superior performance. Meanwhile, safety-critical applications pose more than just resource and performance constraints. In particular, predictions must not be overly confident, i.e., provide properly calibrated uncertainty estimations (proper uncertainty calibration*), and CNNs must be robust against corruptions like naturally occurring input perturbations (*natural corruption robustness*). This work investigates the important trade-off between uncertainty calibration, natural corruption robustness, and performance for current state-of-research post-hoc CNN pruning techniques in the context of image classification tasks. Our study reveals that post-hoc pruning substantially improves the model's uncertainty calibration, performance, and natural corruption robustness, sparking hope for safe and robust embedded CNNs. Furthermore, uncertainty calibration and natural corruption robustness are not mutually exclusive targets under pruning, as evidenced by the improved safety aspects obtained by post-hoc unstructured pruning with increasing compression.*

1. Introduction

In the realm of computer vision, Convolutional Neural Networks (CNNs) have emerged as a dominant paradigm, demonstrating remarkable success in diverse applications, including image classification [58], object detection [61] and video analysis [51]. However, due to their extensive parameter count [7], these architectures demand substantial storage and computational resources, posing limitations on deployment in embedded systems. To address this issue, pruning methods have been introduced that reduce the number of parameters, effectively compress the network, and decrease computational complexity [11, 40]. The primary strategy involves identifying and eliminating the least important network components while preserving desired performance measures, such as accuracy. The pruning landscape consists of two main categories [34]: unstructured and structured pruning. Unstructured pruning deals with the individual weights of a network, resulting in the development of sparse models. In contrast, structured pruning typically eliminates complete channels, filters, or layers. Additionally, concerning the timing of pruning during the training of a neural network, pruning algorithms can be categorized into two groups: post-hoc and ante-hoc pruning [60]. Post-hoc pruning algorithms [18, 33, 35] operate on pre-trained models, utilizing knowledge from the initial training to selectively remove parameters based on criteria. In contrast, ante-hoc pruning algorithms [9, 32] explore effective model architectures during the pre-training phase. Of particular interest for this paper is post-hoc pruning, i.e., pruning of an already trained CNN. In contrast to ante-hoc pruning algorithms that need intervention during training, this allows the distribution of training tasks and integration to different teams or suppliers. Irrespective of the diverse pruning methodologies, pruning algorithms primarily focus on assessing post-pruning accuracy and inference time. Nevertheless, these metrics alone do not provide a comprehensive understanding of other consequences of pruning, such as the impact on model robustness and reliability [13, 28].

In real-world applications, CNN architectures must be robust against perturbations of the inputs [46, 56], such as out-of-distribution data [57]. As of the categorization in [46, 56], common robustness challenges include adversarial attacks, concept drift, and covariate shift. In the case of adversarial attacks, intentional manipulations aim to deceive the model [37]. Concept drift refers to a task-specific output label distribution change due to, e.g., novel objects [55]. Hence, for assessments of architecture-specific robustness, task-agnostic covariate shifts are most relevant, which means changes in the input data distribution over

time due to perturbations such as sensor noise, varying weather conditions, and so on [2].

Several studies demonstrate the vulnerability of CNNs to typical data distortions, compared to human performance [10]. For example, in assessments of image recognition models, CNNs were found to be notably susceptible to covariant shift such as blurring and Gaussian noise [8]. The absence of invariance to slight translations in multiple CNNs was also identified by [3]. Even worse, when it comes to real-world conditions, Von Bernuth et al. [52] report a significant performance degradation of an object detection CNN when evaluated on weather-corrupted data. Meanwhile, state-of-the-art methods for enhancing the robustness of CNNs typically rely on large models [24], impractical for resource-constrained settings. Pruning resolves this issue by reducing memory and computational demands. There has been extensive research on the robustness of pruned models against adversarial attacks [12, 14, 48]; however, adversarial robustness does not necessarily extend to other corruptions like covariate shifts. Therefore, the trade-offs related to natural corruption robustness in pruned networks remain undetermined.

Moreover, while deep learning research focuses on improving the accuracy of networks, less attention has been given to the reliability of the networks. Model uncertainty calibration represents the degree of correspondence between the model output confidences and their actual probability of correctness [39], ensures well-calibrated confidences as a reliable indicator of output trustworthiness. In recent years, a growing body of research has highlighted a respective trend: despite accuracy advancements, modern neural networks often exhibit poor uncertainty calibration [15]. However, in real-world safety-relevant applications, classification networks must be both accurate and well-calibrated to trigger appropriate safety measures during high uncertainty [4, 56]. Pruning introduces structural changes and weight reductions in neural networks, potentially affecting their predictive capabilities and confidence estimates [59]. Despite this, the impact of different post-hoc pruning approaches on uncertainty calibration, including the trade-off between compression and calibration, enjoyed little attention so far.

Since state-of-the-art post-hoc pruning techniques purely consider CNN accuracy for guiding the compression, it is not obvious that pruning will preserve any other safety targets; the structural changes might even result in degradation thereof, rendering pruning inapplicable to safety-relevant applications. For this reason, in this paper, we systematically investigate the uncertainty calibration error and natural corruption robustness of post-hoc pruned CNNs under increasing compression compared to the original network for image classification task. To the best of our knowledge, this is the first research work in the literature benchmarking several popular unstructured and structured post-hoc pruning techniques for image classification while thoroughly and quantitatively studying their uncertainty calibration and natural corruption robustness trade-offs. It provides a detailed understanding of the foremost positive influence of post-hoc pruning on safety properties. Our key findings are:

1. Post-hoc pruning consistently improves the uncertainty calibration compared to their unpruned counterparts; post-hoc unstructured pruned models exhibit substantial improvements in calibration, whereas post-hoc structured pruned models exhibit improved uncertainty calibration performance up to a specific degree of compression.
2. Post-hoc pruning has no negative impact on natural corruption robustness.
3. Post-hoc pruning does not affect uncertainty calibration in the presence of natural corruption.

2. Related Work

2.1. Safety Metrics

When assessing CNNs, most works concentrate on standard performance metrics such as accuracy and F1 score (for classification) or mean average precision (for object detection). To ensure the safety of CNNs, one needs to consider their technology-specific insufficiencies. Sämman et al. [43] and Schwalbe et al. [46] categorized them into: lack of generalized *performance*; incorrect internal *logic*; lack of *robustness* against perturbations that do not change the semantic content of the input; lack of *efficiency* for the respective hardware, usually correlated with the CNN size; and CNN *opaqueness*, which, however, is hard to quantify across use-cases objectively [44, 62]. This catalogue was extended to consider incorrect *uncertainty* estimate outputs separately measured by variants of the calibration error metric - expected, average, and maximum calibration error [15]. Some safety metrics found in the literature are specifically tailored to the safety needs as well as potential logical issues tied to a use-case [45, 56]. Examples are consistency with logical constraints [47] or standard detection accuracy weighted by safety-relevance of objects [6]. However, such metrics require a concrete reference system and are hardly generalized across use cases. Robustness against perturbations, however, gives rise to a rich set of application-agnostic metrics. As of [46] and [56], these are divided into two classes: adversarial and natural corruption robustness. The first considers robustness against adversarial attacks, i.e., targeted and maliciously crafted changes to the input [5, 37, 54], and sets focus on security. On the other side, natural corruption robustness considers naturally occurring non-semantic corruptions of inputs, such as noise and translations from sensor degradation, adverse weather conditions like fog and rain [46]. Percentage change in accuracy, the

relative change in failure rate, mean performance under corruption, and mean corruption error are the most popular metrics for both robustness against adversarial attacks and natural corruption robustness [24, 37]. Our work focuses on *expected calibration error* and *mean performance under corruption* as the safety metrics for uncertainty calibration and natural corruption robustness, respectively. The aforementioned safety measures are explored mostly for vanilla networks without considering model compression.

2.2. Model Compression

In order to leverage the power of large state-of-the-art computer vision CNNs in resource-constrained environments, compression techniques must be employed to reduce the number of parameters and, thus, computation operations and memory of the deployed CNN. As of He et al. [22], typical and complementary means of compression are quantization, i.e., compression of the weight value representation [42], and CNN pruning. Pruning started as early as the 1980s [41], and, unlike hardware-dependent quantization, aims to remove parameters on an architectural level. Following [34]; pruning methods can be differentiated by their application time (dynamic during operation versus static before deployment), their pruning criterion, the removed elements, and whether they are structured or unstructured. Operation-time pruning employs decision logic at runtime to dynamically remove computational paths [34], but cannot decrease CNN memory consumption.

Static pruning approaches can be sub-classified according to their instance of interference before deployment: 1. *Post-hoc: Pruning after training* is a three-step process: first, train the initial network to convergence; second, prune redundant parameters based on specific criteria; and finally, retrain the pruned model (fine-tuning) to recover any performance loss incurred during the pruning process [17, 33, 35]. 2. *Ante-hoc:* (a)*Pruning during training:* compared to pruning after training, connections are dynamically deactivated during training based on their importance. But later weights can adapt and potentially be reactivated based on gradient updates [64]. (b)*Pruning before training resp. weight rewinding:* motivated by the Lottery Ticket Hypothesis [9] some recent studies aim to identify a sparsity mask that can be used for weight initialization, e.g., using information from a previous training run, and subsequently train the pruned network from scratch while maintaining the original mask throughout the training process [32, 53]. In this article, we have adopted approach 1., i.e., pruning after training or post-hoc pruning, which not only has the largest body of research [31], but also most practical relevance due to the separation of concerns of model training and integration. In post-hoc pruning, the model can be pruned with different compression ratios without the necessity of initiating training from its initial state. This affords flexibility in modifying compression ratios without undertaking a comprehensive model retraining. Conversely, when employing ante-hoc algorithms for model pruning with diverse compression ratios, it is imperative to initiate model retraining to accommodate the specific compression ratio, contrasting the more adaptable approach offered by post-hoc pruning.

In our work, we consider two fundamental categories of static post-hoc pruning: unstructured pruning and structured pruning. *Unstructured pruning* involves the removal of individual weights which are assigned the least importance for the network functioning, resulting in a sparse network without sacrificing predictive performance [17, 18]. However, since the positions of non-zero weights are irregular and random, the sparse network pruned by unstructured pruning cannot be presented in a *structured* fashion. This means that it cannot lead to compression and speedup without dedicated hardware or libraries [19]. To address this, so-called *structured pruning* typically prunes complete channels, filters, or layers within the CNN architectural structure according to an importance criterion [21, 33, 35]. As a result, the pruned network retains the original convolutional structure without introducing sparsity, and no sparse libraries or specialized hardware are required to realize the benefits of pruning. Since both have practical advantages, we cover and compare these two pruning paradigms in this study.

For ease of notation, pruning refers to post-hoc pruning in the remainder of this paper.

2.3. Pruning and Uncertainty Calibration

There is a limited exploration of the impact of pruning on uncertainty calibration. While a study conducted by Sun et al. [50] indicates that sparsification is beneficial for enhancing the calibration of CNNs, their study specifically focused on various unstructured pruning methods applied to smaller residual networks (ResNet-20, ResNet-32) with lower sparsity levels (up to 20%). Consequently, these investigations lack a comprehensive analysis of diverse pruning methodologies involving larger residual networks and higher sparsity levels. This gap in research underscores the fact that the impact of pruning on uncertainty calibration remains both active and largely unexplored which we aim to tackle here.

2.4. Pruning and Robustness

There is a growing body of literature on the robustness of pruning methods against adversarial attacks, focusing mainly on the improvement of the adversarial robustness of pruned models even without adversarial training [16, 27]. According to these results, pruned models typically do *not* inherit the susceptibility to adversarial attacks observed in the original models. The previously cited studies demonstrate the positive effect of pruning concerning robustness against adversarial attacks. However, this is limited to

adversarial robustness, *not taking into account* robustness against naturally occurring corruptions like fog or random noise from sensor degradation.

Compared to research on adversarial robustness, there is a limited exploration of the impact of pruning on natural corruption robustness. Regarding the relationship between pruning and natural corruption robustness, the study conducted by Hooker et al. [26] revealed that the corruption performance significantly deteriorates when higher pruning ratios are applied for the magnitude pruning method. According to Hoffmann et al. [25], unstructured (*global weight*) pruning preserves more robustness regarding performance under corruption than the structured (L_1*-norm filter*) pruning counterparts. However, such studies lack a comprehensive examination of diverse pruning methodologies, leaving the influence of pruning on natural corruption robustness an active and unexplored research area.

3. Methods

3.1. Pruning

All static pruning methods commonly consist of a *step 1*, in which they determine an importance score for candidate network units, *step 2*, in which this is used to select and remove items with a low score, and an optional last step of fine-tuning the newly obtained pruned network. The decision threshold in the second step influences the final *pruning ratio*, i.e., the percentage of removed parameters, which can serve as a measure for the achieved compression.

As outlined in Section 2.2, static pruning approaches that apply after model training can be divided into *unstructured pruning* of weights, and *structured pruning* of filters or channels in case of CNNs [34]. To evaluate pruning with respect to uncertainty calibration and natural corruption robustness and to compare different pruning paradigms with respect to these aspects, we consider the following three pruning methods, which are commonly used and chosen to cover a broad range of static pruning approaches [30].

3.1.1 Unstructured Weight Pruning

We consider the unstructured pruning strategy from Han et al. [18] for magnitude-based weight pruning, comprising two steps. *Step 1:* globally sort the weights according to their relative importance based on the magnitude of the weights calculated by L_1-norm. *Step 2:* prune $k\%$ of the weights with the lowest importance, where $k\%$ is the pruning ratio.

3.1.2 Structured Filter Pruning

We adopt the L_1-norm-based filter pruning technique by Li et al. [33] as a filter-level structured pruning method. *Step 1:* rank the filters by their L_1-norm value in each convolutional layer. *Step 2:* remove the $k\%$ lowest ranking filters, where $k\%$ approximates the pruning ratio.

3.1.3 Structured Channel Pruning

Additionally, we employ network slimming introduced by Liu et al. [35] as another structured pruning method targeting channel-level sparsity. *Step 1:* During training, impose L_1 sparsity regularization on the channel-wise scaling factors for each channel from batch normalization layers. *Step 2:* Remove those channels with near-zero scaling factors afterwards.

3.2. Safety Metrics

Besides efficiency, the two main application-agnostic safety targets for computer vision CNNs are correct uncertainty calibration and robustness against naturally occurring corruptions [37, 46, 62]. These are quantified by means of expected calibration error and mean performance under corruption on a benchmark corruption dataset, respectively.

3.2.1 Uncertainty Calibration

In the context of employing pruned models in safety-critical applications, ensuring good uncertainty calibration is of high importance, in addition to achieving the desired levels of sharpness of uncertainty. In this paper, we want to investigate whether pruning has an adverse effect on CNN uncertainty calibration. Therefore, the calibration error concept is employed to re-assess the uncertainty calibration properties of the pruned models with respect to the original unpruned model.

Good uncertainty calibration aims to ensure that the predicted confidences correctly represent the actual probability of the correctness of the prediction. Miscalibration is commonly quantified in terms of Expected Calibration Error (ECE) [15]. This is measured by first binning samples of the test set according to their predicted confidence value; then measuring the accuracy for each bin; and lastly, determining the weighted average of the difference between bins' accuracy and mean predicted confidence. This formalizes to

$$ECE = \sum_{m=1}^{M} \frac{B_m}{n} \left| acc(B_m) - conf(B_m) \right|, \qquad (1)$$

where M is the total number of bins into which the predictions are equally grouped, B_m is the number of samples whose prediction confidence falls into the interval $I_m = \left(\frac{m-1}{M}, \frac{m}{M}\right]$ and n is the total number of samples. The accuracy of B_m is defined as:

$$acc(B_m) = \frac{1}{|B_m|} \sum_{i \in B_m} \mathcal{I}(\hat{y}_i = y_i), \qquad (2)$$

where $\hat{y}_i$ and y_i are the predicted and true class labels for sample i, and $\mathcal{I}(\cdot)$ is the indicator function. Finally, the average confidence within bin B_m is defined as:

$$conf(B_m) = \frac{1}{|B_m|} \sum_{i \in B_m} \hat{p}_i, \tag{3}$$

where $\hat{p}_i$ is the confidence for sample i.

3.2.2 Natural Corruption Robustness

Robustness to distribution shift is an important feature of CNNs for real-world applications, where the environmental conditions may vary substantially. Among many forms of distribution shift, one particularly relevant category for computer vision is covariate shift, i.e., input image corruption [2, 46, 56]. Therefore, natural corruption robustness is important when deploying pruned models in safety-critical applications. This paper investigates how pruning influences natural corruption robustness compared to the original unpruned model. This inquiry aims to provide insights into the potential trade-offs and implications of pruning methodologies concerning the network's ability to withstand natural corruption in safety-critical applications.

The natural corruption robustness of a (pruned or unpruned) model is evaluated in terms of *mean performance under corruption*(mPC) [38], which is defined as

$$mPC = \frac{1}{N_c} \sum_{c=1}^{N_c} \frac{1}{N_s} \sum_{s=1}^{N_s} P_{c,s}, \tag{4}$$

where $P_{c,s}$ represents the performance computed on test data corrupted with corruption type c under severity levels s. N_s and N_c denote the number of severity levels and corruptions respectively.

4. Experimental Setup

The experimental setup for benchmarking the pruning methods from Section 3.1 against the safety metrics from Section 3.2 and accuracy is detailed below.

Concretely, the investigated research questions are:

(i) Do increasing pruning ratios affect any of
   a. uncertainty calibration (see Section 5.1);
   b. natural corruption robustness (see Section 5.2); or
   c. uncertainty calibration when challenged with natural corruptions (see Section 5.3);

(ii) Is there a difference between structured and unstructured pruning in any of the above cases?

4.1. Datasets & Models

Within the network pruning literature, CIFAR-10 stands as the established benchmark dataset, and VGG, ResNet serve as the prevalent network architectures. Our assessment of the three pruning methods aligns with the target models and dataset pairs presented in the original paper, ensuring the comparability of our results.

CIFAR-10: A common benchmark dataset for image classification is CIFAR-10 [29]. It contains 60,000 (50,000 training and 10,000 test images) colour images of 32×32 resolution in 10 classes, with an equal distribution of 6,000 images per class. We use the CIFAR-10 training data as in-distribution data (clean data) to train the CNNs and to fine-tune the pruned models. The test split is used to determine the safety metrics of unpruned and pruned models on clean data.

CIFAR-10-C: This dataset is constructed by synthetically corrupting the original CIFAR test sets [23]. It consists of 15 types of corruption, each further categorized into five distinct severity levels, containing 50,000 images for each type of corruption. The corruptions cover four categories: noise, blur, weather effects, and digital transforms. In this paper, CIFAR-10-C is used as naturally corrupted data during testing to check the natural corruption robustness of original and pruned CNN models [39].

Models: VGG networks, introduced by Simonyan and Zisserman [49], leverage a deep architecture with small convolutional filters, showcasing robust performance in image classification task [1]. Residual network backbones, pioneered by He et al. [20], prove highly effective in mitigating the vanishing gradient problem, enabling the training of extremely deep networks and achieving state-of-the-art results in image classification tasks [1].

To ensure comparability between unstructured and structured pruning, we choose VGG-19 and PreResNet-110 for weight pruning, VGG-16 and ResNet-110 for filter pruning and VGG-19 and ResNet-164 for the more rigorous channel pruning. This setup is able to reproduce accuracy comparable to the reported results of the baseline models from the original works (for accuracy on clean test data, see Figure 1).

4.2. Training Configuration

We adopt the implementation and hyperparameters for weight pruning, filter pruning, and channel pruning from the publicly available codebase by Liu et al. [36], demonstrating comparable results to the original works. Using a stochastic gradient descent optimizer, the original models are trained for 160 epochs with a batch size of 64. An exponentially decreasing learning rate is applied, starting at 0.1 for epochs [1, 80), 0.01 for epochs [80, 120), and finally 0.001 until epoch 160. Simple data augmentation involving random crop and random horizontal flip is used on training images as a standard means to foster natural corruption robustness in the original models. For fine-tuning the pruned models, we use a constant learning rate set to the last one used for training the original model (0.001) and apply this

for 40 epochs.

4.3. Pruning Ratio Selection

The pruning ratios vary among the selected pruning methods, each characterized as follows:

Magnitude-based weight pruning: The pruning ratio, defined as the percentage of parameters pruned within the convolutional weights of convolution layers, establishes the pruning threshold. It determines which weights are set to zero based on their magnitudes relative to the specified threshold value.

Filter Pruning: VGG-16 on CIFAR-10 comprises 13 convolutional layers and 2 fully connected layers. Pruning 512-feature map layers, as reported in [33], maintain accuracy due to filters' limited spatial connections on small feature maps. Despite extensive pruning in the first layer, the remaining filters outnumber input channels. However, excessive pruning in the second layer risks losing vital information. To maintain compatibility with the original implementation, our approach selectively prunes layers 1 and 8 to 13.

ResNets designed for CIFAR-10 comprise three stages of residual blocks, handling feature maps of sizes 32×32, 16×16, and 8×8. Each stage maintains an identical number of residual blocks. Deeper layers exhibit increased sensitivity to pruning compared to earlier stages, as noted in [33]. Specifically targeting the first layer of the residual block, our approach ensures compatibility with the original implementation by exclusively pruning the first layer of each residual block within the first stage of the network. The pruning ratio reflects the percentage of filters pruned for all first layers in the first stage.

Channel Pruning: We adopt the approach outlined in [35] by utilizing a universal pruning threshold applied consistently across all layers. This threshold is established based on a percentile value among all scaling factors. For instance, we prune channels by selecting those with lower scaling factors, which is achieved by setting the percentile threshold accordingly.

4.4. Evaluation Metrics

Performance: We use classification accuracy to measure the performance of original and pruned models on the clean dataset.

Natural Corruption Robustness: We report mean accuracy under corruption as a performance metric (mPC, see Section 3.2.2) overall corruption types for each severity level.

Uncertainty Calibration: To estimate the miscalibration of the original and pruned model on the clean and corrupted datasets, we use ECE (see Section 3.2.1) using equal-mass binning with ten bins.

Compression: To examine the impact of pruning on the other safety targets, we benchmark the three different pruning methods from Section 3.1. The resulting compression is measured in terms of their respective pruning ratio (Section 4.3), which is sampled at a rate of 10% from values 0-70%, where 0% indicates the original network (unpruned).

5. Results

5.1. Does pruning affect uncertainty calibration?

To examine the impact of pruning on network uncertainty calibration, we compared ECE and accuracy under different pruning ratios for the three selected pruning techniques. The results are shown in Figure 1 for weight, filter, and channel pruning, respectively (we conduct each experiment three times and report mean $\pm$ std).

The results suggest that *even high pruning ratios do not impact the uncertainty calibration* compared to that of the original unpruned model. In unstructured (weight) pruning, the calibration error for all pruning ratios is less than the calibration error of the original unpruned model. Hence, unstructured pruning can even enhance the uncertainty calibration of the VGG-19 and ResNet-110 model. Whereas, in filter and channel pruning, the calibration error for pruned models up to a certain pruning ratio (filter pruning: 50% on VGG-16 and 50% on ResNet-110, channel pruning: 40% on VGG-19 and 50% on ResNet-164) is less than or similar to the calibration error of the original unpruned model. After that, the calibration errors of pruned models do not increase substantially with respect to the calibration error of the unpruned model.

5.2. Does pruning affect natural corruption robustness?

To answer this question, the choice of pruning methods and ratio are kept similar, as mentioned in Section 5.1, but naturally corrupted data is considered as test data. Here, mPC is measured for unpruned and pruned models for the selected pruning methods from Section 3.1. mPC is calculated separately for each of the five corruption severity levels from CIFAR-10-C, each pruning ratio step, and each pruning method. Figure 2 illustrates how natural corruption robustness is influenced by pruning for different severity levels for weight, filter, and channel pruning, respectively.

The observation implies that the *natural corruption robustness of weight pruned and filter pruned models, as assessed through mPC, is better or similar compared to the original unpruned model*. Notably, the robustness against natural input corruption remains unaffected by the weight pruning of VGG-19, ResNet-110, filter pruning of VGG-16, ResNet-110, and channel pruning of VGG-19 across all severity levels. In contrast, for channel pruning, the mPC of ResNet-164 starts to degrade from ca. 60% pruning ratio for all corruption levels. Consequently, the robustness expe-

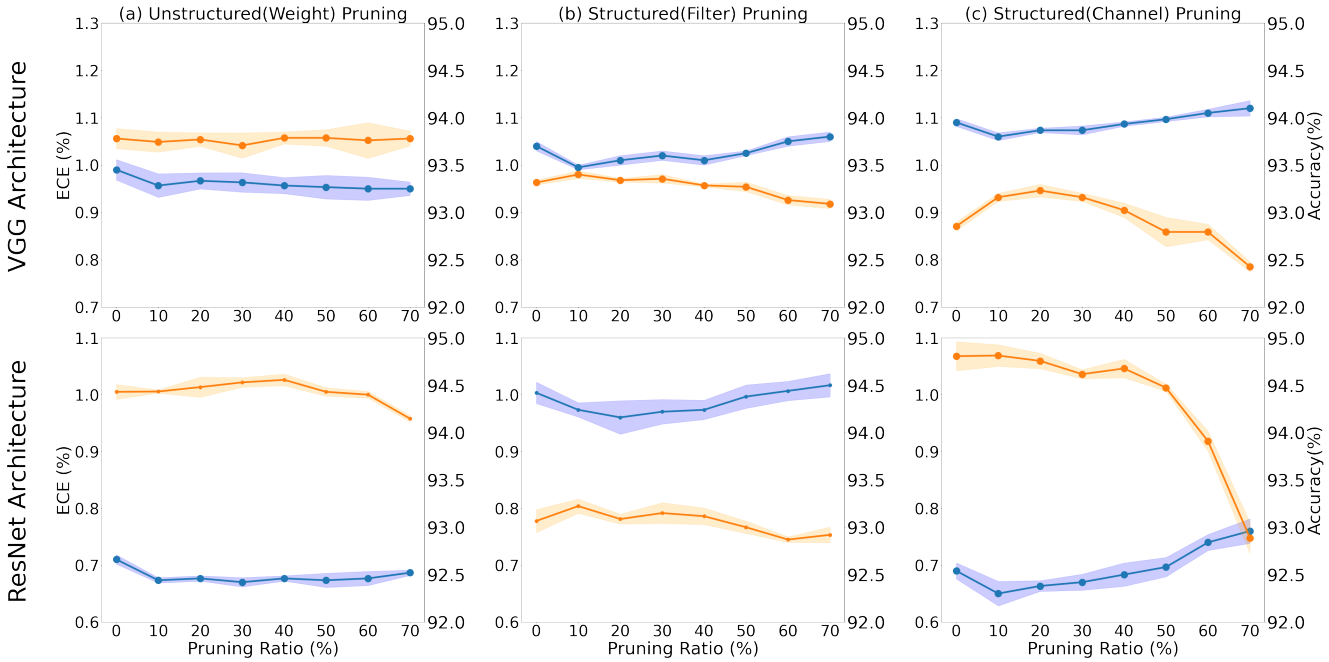

Figure 1. Comparing development of *ECE* (↓) (blue lines, left y-axis scaling) and accuracy (↑) (orange lines, right y-axis scaling) under increasing pruning ratios

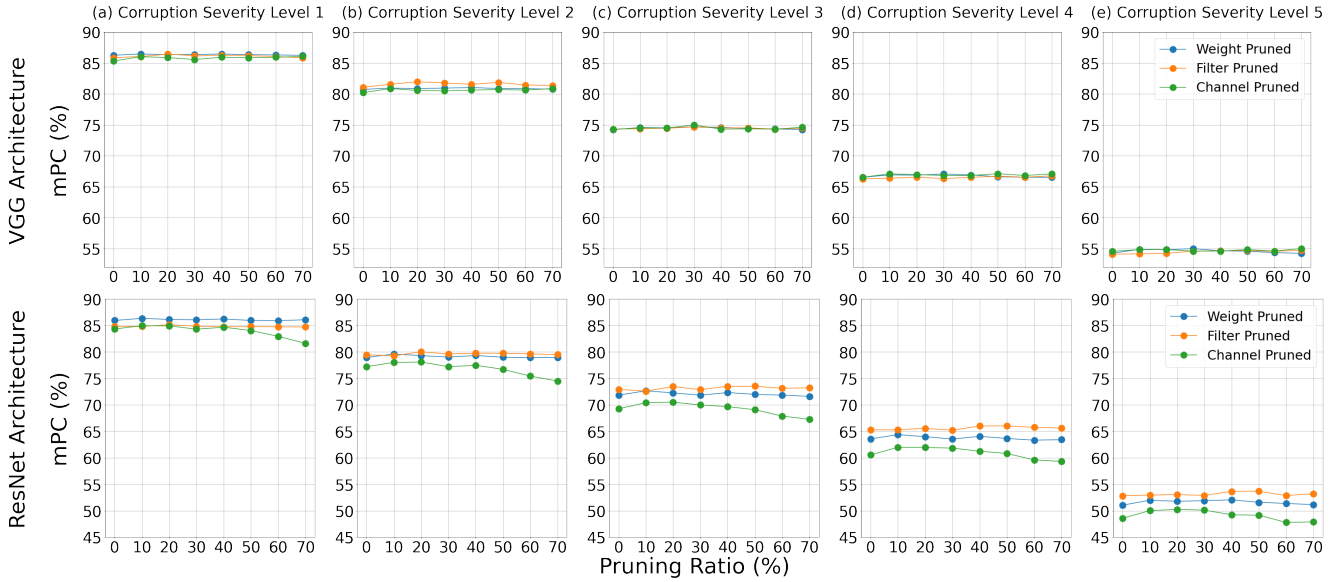

Figure 2. Development of mPC (↑) for increasing pruning ratios for five severity levels of corruption

riences deterioration with higher pruning rates for channel pruning in the ResNet-164 model across all severity levels. This suggests that information needed to compensate for corruption is highly distributed over channels, i.e., wider networks (in terms of the number of channels) might have better chances of achieving zero-shot natural corruption robustness than narrow ones.

While the pruning ratio seems to play a negligible role in accuracy in the presence of corruption, one should, however, note the severe drop in accuracy for increasing corruption severity levels (from more than 90% without corruption in Figure 1 down to less than 55%). This attests to the generally weak overall robustness of CNNs against strong natural corruption.

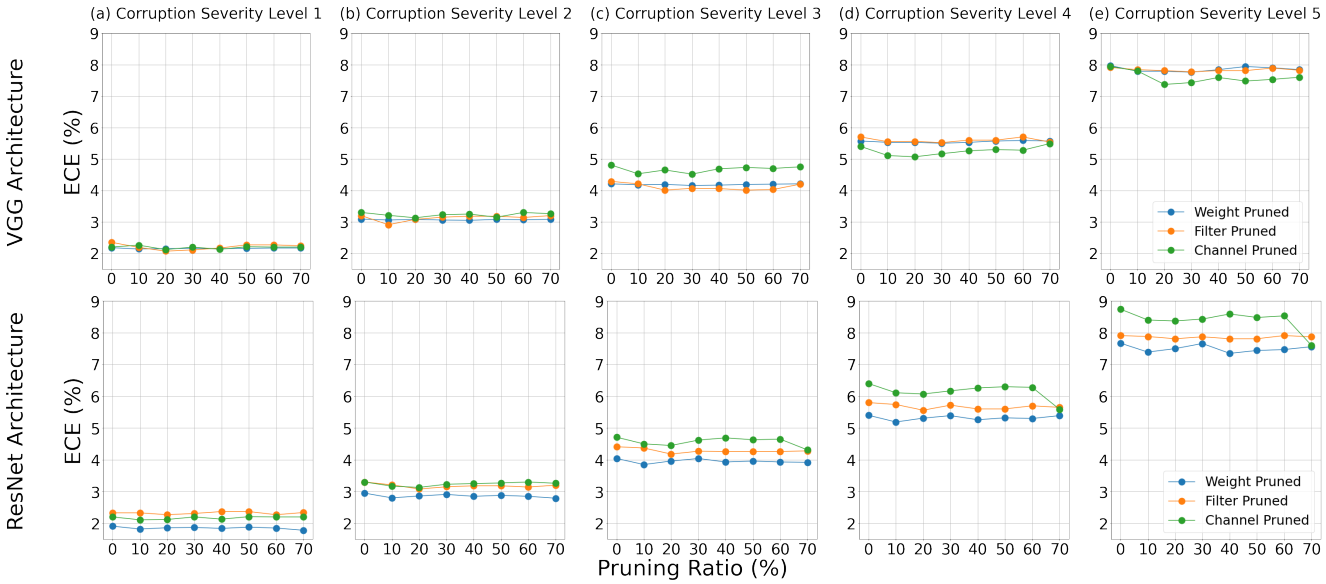

Figure 3. Development of ECE (↓) for increasing pruning ratio for five severity levels of corruption

5.3. Does pruning impact uncertainty calibration in the presence of natural corruption?

Here, ECE is measured for unpruned and pruned models using the different pruning methods from above in the presence of natural corruption of different severity levels. Figure 3 illustrates how the uncertainty calibration is influenced by pruning in the presence of natural corruption for weight, filter, and channel pruning, respectively.

The results demonstrate that *pruning does not negatively impact the model uncertainty calibration compared to the original unpruned model, even when additionally confronted with natural corruptions*. In weight, filter, and channel pruning, the measured calibration error for all pruned models at different pruning ratios is similar to or less than the calibration error of the original unpruned VGG-19, VGG-16, ResNet-110 and ResNet-164 models for all five severity levels of corruption. Nevertheless, as for mPC, ECE increases rapidly with increasing severity levels and worse mPC, reaching more than 400% increase of ECE for the highest level. Hence, natural corruptions seem not only to pose a challenge to robust accuracy but also to trustworthiness in terms of uncertainty calibration.

6. Conclusion

For safety-critical computer vision applications, model efficiency, proper uncertainty calibration, and natural corruption robustness are—and will be—the key desirables. This work, for the first time, investigated whether popular post-hoc pruning as a means for model compression conflicts with the other two safety targets. Our benchmark with a standard setup of model, dataset, and post-hoc pruning methods provided promising insights into this: we could *not* find a negative effect of pruning on natural corruption robustness and uncertainty calibration; calibration was not even affected by pruning when challenged with naturally corrupted inputs. Our considered post-hoc unstructured pruning method showed a consistently positive effect on uncertainty calibration even when pruning up to 70%. While our results on typical image classification backends do not yet cover the whole spectrum of computer vision tasks and architectures, they raise hope that accuracy-driven pruning does not contradict but even enhances other safety targets. Future work includes extending our investigation to costly experimental setup tasks like object detection and semantic segmentation.

Further research in the realm of safe pruning could explore tailored safety objectives, such as the impact or chances of pruning for interpretability and out-of-distribution generalization and detection capabilities [63]. Such endeavours hold promise for advancing safety metrics in pruning practices.

We aim to raise awareness of cross-discipline safety challenges in model compression, uncertainty calibration, and robustness, serving as an initial step towards exploring common solutions.

7. Acknowledgement

The research leading to these results is funded by the German Federal Ministry of Education and Research (BMBF) within the project CeCaS (“Central Car Server-Supercomputing”). The authors would like to thank the consortium for the successful cooperation.

References

[1] Laith Alzubaidi, Jinglan Zhang, Amjad J Humaidi, Ayad Al-Dujaili, Ye Duan, Omran Al-Shamma, José Santamaría, Mohammed A Fadhel, Muthana Al-Amidie, and Laith Farhan. Review of deep learning: Concepts, cnn architectures, challenges, applications, future directions. *Journal of big Data*, 8:1–74, 2021. 5

[2] Dario Amodei, Chris Olah, Jacob Steinhardt, Paul F. Christiano, John Schulman, and Dan Mané. Concrete problems in ai safety. *CoRR*, abs/1606.06565, 2016. 2, 5

[3] Aharon Azulay and Yair Weiss. Why do deep convolutional networks generalize so poorly to small image transformations? *arXiv preprint arXiv:1805.12177*, 2018. 2

[4] Mariusz Bojarski, Davide Del Testa, Daniel Dworakowski, Bernhard Firner, Beat Flepp, Prasoon Goyal, Lawrence D Jackel, Mathew Monfort, Urs Muller, Jiakai Zhang, et al. End to end learning for self-driving cars. *arXiv preprint arXiv:1604.07316*, 2016. 2

[5] Nicholas Carlini and David Wagner. Towards evaluating the robustness of neural networks. In *2017 IEEE Symposium on Security and Privacy (SP)*, pages 39–57, 2017. 2

[6] Chih-Hong Cheng, Georg Nührenberg, Chung-Hao Huang, Harald Ruess, and Hirotoshi Yasuoka. Towards dependability metrics for neural networks. In *16th ACM/IEEE Int. Conf. Formal Methods and Models for System Design*, pages 43–46. IEEE, 2018. 2

[7] Emily L Denton, Wojciech Zaremba, Joan Bruna, Yann LeCun, and Rob Fergus. Exploiting linear structure within convolutional networks for efficient evaluation. *Advances in neural information processing systems*, 27, 2014. 1

[8] Samuel Dodge and Lina Karam. Understanding how image quality affects deep neural networks. In *2016 eighth international conference on quality of multimedia experience (QoMEX)*, pages 1–6. IEEE, 2016. 2

[9] Jonathan Frankle and Michael Carbin. The lottery ticket hypothesis: Finding sparse, trainable neural networks. *arXiv preprint arXiv:1803.03635*, 2018. 1, 3

[10] Robert Geirhos, Carlos RM Temme, Jonas Rauber, Heiko H Schütt, Matthias Bethge, and Felix A Wichmann. Generalisation in humans and deep neural networks. *Advances in neural information processing systems*, 31, 2018. 2

[11] Sayan Ghosh, Karthik Prasad, Xiaoliang Dai, Peizhao Zhang, Bichen Wu, Graham Cormode, and Peter Vajda. Pruning compact convnets for efficient inference. *arXiv preprint arXiv:2301.04502*, 2023. 1

[12] Micah Goldblum, Liam Fowl, Soheil Feizi, and Tom Goldstein. Adversarially robust distillation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, pages 3996–4003, 2020. 2

[13] Brunno F Goldstein, Sudarshan Srinivasan, Dipankar Das, Kunal Banerjee, Leandro Santiago, Victor C Ferreira, Alexandre S Nery, Sandip Kundu, and Felipe MG França. Reliability evaluation of compressed deep learning models. In *2020 IEEE 11th Latin American Symposium on Circuits & Systems (LASCAS)*, pages 1–5. IEEE, 2020. 1

[14] Shupeng Gui, Haotao Wang, Haichuan Yang, Chen Yu, Zhangyang Wang, and Ji Liu. Model compression with adversarial robustness: A unified optimization framework. *Advances in Neural Information Processing Systems*, 32, 2019. 2

[15] Chuan Guo, Geoff Pleiss, Yu Sun, and Kilian Q. Weinberger. On calibration of modern neural networks. In *Proc. 2017 Int. Conf. Machine Learning*, pages 1321–1330. PMLR, 2017. 2, 4

[16] Yiwen Guo, Chao Zhang, Changshui Zhang, and Yurong Chen. Sparse dnns with improved adversarial robustness. *Advances in neural information processing systems*, 31, 2018. 3

[17] Song Han, Huizi Mao, and William J Dally. Deep compression: Compressing deep neural networks with pruning, trained quantization and huffman coding. *arXiv preprint arXiv:1510.00149*, 2015. 3

[18] Song Han, Jeff Pool, John Tran, and William Dally. Learning both weights and connections for efficient neural network. *Advances in neural information processing systems*, 28, 2015. 1, 3, 4

[19] Song Han, Xingyu Liu, Huizi Mao, Jing Pu, Ardavan Pedram, Mark A Horowitz, and William J Dally. Eie: Efficient inference engine on compressed deep neural network. *ACM SIGARCH Computer Architecture News*, 44(3):243–254, 2016. 3

[20] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proc. 2016 IEEE Conf. Comput. Vision and Pattern Recognition*, pages 770–778, 2016. 5

[21] Yihui He, Xiangyu Zhang, and Jian Sun. Channel pruning for accelerating very deep neural networks. In *Proceedings of the IEEE international conference on computer vision*, pages 1389–1397, 2017. 3

[22] Yihui He, Ji Lin, Zhijian Liu, Hanrui Wang, Li-Jia Li, and Song Han. Amc: Automl for model compression and acceleration on mobile devices. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 784–800, 2018. 3

[23] Dan Hendrycks and Thomas Dietterich. Benchmarking neural network robustness to common corruptions and perturbations. *arXiv preprint arXiv:1903.12261*, 2019. 5

[24] Dan Hendrycks and Thomas G Dietterich. Benchmarking neural network robustness to common corruptions and surface variations. *arXiv preprint arXiv:1807.01697*, 2018. 2, 3

[25] Jasper Hoffmann, Shashank Agnihotri, Tonmoy Saikia, and Thomas Brox. Towards improving robustness of compressed cnns. In *ICML Workshop on Uncertainty and Robustness in Deep Learning (UDL)*, 2021. 4

[26] Sara Hooker, Aaron Courville, Gregory Clark, Yann Dauphin, and Andrea Frome. What do compressed deep neural networks forget? *arXiv preprint arXiv:1911.05248*, 2019. 4

[27] Artur Jordao and Hélio Pedrini. On the effect of pruning on adversarial robustness. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 1–11, 2021. 3

[28] Sawinder Kaur, Ferdinando Fioretto, and Asif Salekin. Deadwooding: Robust global pruning for deep neural networks. *arXiv preprint arXiv:2202.05226*, 2022. 1

[29] Alex Krizhevsky, Geoffrey Hinton, et al. Learning multiple layers of features from tiny images. 2009. 5

[30] Duong Hoang Le and Binh-Son Hua. Network pruning that matters: A case study on retraining variants. In *International Conference on Learning Representations*, 2020. 4

[31] Duong H Le and Binh-Son Hua. Network pruning that matters: A case study on retraining variants. *arXiv preprint arXiv:2105.03193*, 2021. 3

[32] Namhoon Lee, Thalaiyasingam Ajanthan, and Philip HS Torr. Snip: Single-shot network pruning based on connection sensitivity. *arXiv preprint arXiv:1810.02340*, 2018. 1, 3

[33] Hao Li, Asim Kadav, Igor Durdanovic, Hanan Samet, and Hans Peter Graf. Pruning filters for efficient convnets. *arXiv preprint arXiv:1608.08710*, 2016. 1, 3, 4, 6

[34] Tailin Liang, John Glossner, Lei Wang, Shaobo Shi, and Xiaotong Zhang. Pruning and quantization for deep neural network acceleration: A survey. *Neurocomputing*, 461:370–403, 2021. 1, 3, 4

[35] Zhuang Liu, Jianguo Li, Zhiqiang Shen, Gao Huang, Shoumeng Yan, and Changshui Zhang. Learning efficient convolutional networks through network slimming. In *Proceedings of the IEEE international conference on computer vision*, pages 2736–2744, 2017. 1, 3, 4, 6

[36] Zhuang Liu, Mingjie Sun, Tinghui Zhou, Gao Huang, and Trevor Darrell. Rethinking the value of network pruning. *arXiv preprint arXiv:1810.05270*, 2018. 5

[37] Charles Meyers, Tommy Löfstedt, and Erik Elmroth. Safety-critical computer vision: An empirical survey of adversarial evasion attacks and defenses on computer vision systems. *Artificial Intelligence Review*, 2023. 1, 2, 3, 4

[38] Claudio Michaelis, Benjamin Mitzkus, Robert Geirhos, Evgenia Rusak, Oliver Bringmann, Alexander S Ecker, Matthias Bethge, and Wieland Brendel. Benchmarking robustness in object detection: Autonomous driving when winter is coming. *arXiv preprint arXiv:1907.07484*, 2019. 5

[39] Matthias Minderer, Josip Djolonga, Rob Romijnders, Frances Hubis, Xiaohua Zhai, Neil Houlsby, Dustin Tran, and Mario Lucic. Revisiting the calibration of modern neural networks. *Advances in Neural Information Processing Systems*, 34:15682–15694, 2021. 2, 5

[40] Pavlo Molchanov, Arun Mallya, Stephen Tyree, Iuri Frosio, and Jan Kautz. Importance estimation for neural network pruning. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 11264–11272, 2019. 1

[41] R. Reed. Pruning algorithms-a survey. *IEEE Transactions on Neural Networks*, 4(5):740–747, 1993. 3

[42] Babak Rokh, Ali Azarpeyvand, and Alireza Khanteymoori. A comprehensive survey on model quantization for deep neural networks in image classification. *ACM Transactions on Intelligent Systems and Technology*, 2023. 3

[43] Timo Sämann, Peter Schlicht, and Fabian Hüger. Strategy to increase the safety of a dnn-based perception for had systems. *CoRR*, abs/2002.08935, 2020. 2

[44] Gesina Schwalbe and Bettina Finzel. A comprehensive taxonomy for explainable artificial intelligence: A systematic survey of surveys on methods and concepts. *Data Mining and Knowledge Discovery*, 2023. 2

[45] Gesina Schwalbe and Martin Schels. A survey on methods for the safety assurance of machine learning based systems. In *Proc. 10th European Congress Embedded Real Time Software and Systems*, 2020. 2

[46] Gesina Schwalbe, Bernhard Knie, Timo Sämann, Timo Dobberphul, Lydia Gauerhof, Shervin Raafatnia, and Vittorio Rocco. Structuring the safety argumentation for deep neural network based perception in automotive applications. In *Computer Safety, Reliability, and Security. SAFECOMP 2020 Workshops*, pages 383–394. Springer International Publishing, 2020. 1, 2, 4, 5

[47] Gesina Schwalbe, Christian Wirth, and Ute Schmid. Enabling verification of deep neural networks in perception tasks using fuzzy logic and concept embeddings, 2022. 2

[48] Vikash Sehwag, Shiqi Wang, Prateek Mittal, and Suman Jana. Hydra: Pruning adversarially robust neural networks. *Advances in Neural Information Processing Systems*, 33: 19655–19666, 2020. 2

[49] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014. 5

[50] Yan Sun, Wenjun Xiong, and Faming Liang. Sparse deep learning: A new framework immune to local traps and miscalibration. *Advances in Neural Information Processing Systems*, 34:22301–22312, 2021. 3

[51] Ruben Villegas, Arkanath Pathak, Harini Kannan, Dumitru Erhan, Quoc V Le, and Honglak Lee. High fidelity video prediction with large stochastic recurrent neural networks. *Advances in Neural Information Processing Systems*, 32, 2019. 1

[52] Alexander Von Bernuth, Georg Volk, and Oliver Bringmann. Simulating photo-realistic snow and fog on existing images for enhanced cnn training and evaluation. In *2019 IEEE Intelligent Transportation Systems Conference (ITSC)*, pages 41–46. IEEE, 2019. 2

[53] Chaoqi Wang, Guodong Zhang, and Roger Grosse. Picking winning tickets before training by preserving gradient flow. *arXiv preprint arXiv:2002.07376*, 2020. 3

[54] Xingxing Wei, Bangzheng Pu, Jiefan Lu, and Baoyuan Wu. Visually adversarial attacks and defenses in the physical world: A survey, 2023. 2

[55] Gerhard Widmer and Miroslav Kubat. Learning in the presence of concept drift and hidden contexts. *Machine Learning*, 23(1):69–101, 1996. 1

[56] Oliver Willers, Sebastian Sudholt, Shervin Raafatnia, and Stephanie Abrecht. Safety concerns and mitigation approaches regarding the use of deep learning in safety-critical perception tasks. In *Computer Safety, Reliability, and Security. SAFECOMP 2020 Workshops*, pages 336–350. Springer International Publishing, 2020. 1, 2, 5

[57] Jingkang Yang, Kaiyang Zhou, Yixuan Li, and Ziwei Liu. Generalized out-of-distribution detection: A survey, 2022. 1

[58] Zhaohui Yang, Yunhe Wang, Xinghao Chen, Boxin Shi, Chao Xu, Chunjing Xu, Qi Tian, and Chang Xu. Cars: Continuous evolution for efficient neural architecture search. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 1829–1838, 2020. 1

[59] Xiaoyong Yuan and Lan Zhang. Membership inference attacks and defenses in neural network pruning. In *31st USENIX Security Symposium (USENIX Security 22)*, pages 4561–4578, 2022. 2

[60] Li Yue, Zhao Weibin, and Shang Lin. Really should we pruning after model be totally trained? pruning based on a small amount of training. *arXiv preprint arXiv:1901.08455*, 2019. 1

[61] Zhong-Qiu Zhao, Peng Zheng, Shou-tao Xu, and Xindong Wu. Object detection with deep learning: A review. *IEEE transactions on neural networks and learning systems*, 30 (11):3212–3232, 2019. 1

[62] Jianlong Zhou, Amir H. Gandomi, Fang Chen, and Andreas Holzinger. Evaluating the quality of machine learning explanations: A survey on methods and metrics. *Electronics*, 10 (5):593, 2021. 2, 4

[63] Jianing Zhu, Hengzhuang Li, Jiangchao Yao, Tongliang Liu, Jianliang Xu, and Bo Han. Unleashing mask: Explore the intrinsic out-of-distribution detection capability. *arXiv preprint arXiv:2306.03715*, 2023. 8

[64] Michael Zhu and Suyog Gupta. To prune, or not to prune: exploring the efficacy of pruning for model compression. *arXiv preprint arXiv:1710.01878*, 2017. 3