Benchmarking Adversarial Robustness of Compressed Deep Learning Models

Brijesh Vora [*1] **Kartik Patwari** [*2] **Syed Mahbub Hafiz** [1] **Zubair Shafiq** [1] **Chen-Nee Chauh** [2]

Abstract

The increasing size of Deep Neural Networks (DNNs) poses a pressing need for model compression, particularly when employed on resource-constrained devices. Concurrently, the susceptibility of DNNs to adversarial attacks presents another significant hurdle. Despite substantial research on both model compression and adversarial robustness, their joint examination remains underexplored. Our study bridges this gap, seeking to understand the effect of adversarial inputs crafted for base models on their pruned versions. To examine this relationship, we have developed a comprehensive benchmark across diverse adversarial attacks and popular DNN models. We uniquely focus on models not previously exposed to adversarial training and apply pruning schemes optimized for accuracy and performance. Our findings reveal that while the benefits of pruning – enhanced generalizability, compression, and faster inference times – are preserved, adversarial robustness remains comparable to the base model. This suggests that model compression while offering its unique advantages, does not undermine adversarial robustness.

1. Introduction

Deep neural networks have continued to exhibit impressive performance in various machine learning applications, including computer vision, natural language processing, object detection, and so on. However, the deployment of these networks on resource-limited devices presents a challenge due to their substantial memory and computational requirements (Chen et al., 2020a). A potential solution to this is neural network compression via pruning (Zhao et al., 2019), which aims to decrease size by identifying and eliminating connections that contribute less to the network's overall performance – pruning effectively reduces the number of parameters and computations required during inference. This compression technique optimizes the network's efficiency by focusing resources on the most critical connections, thereby enhancing its computational speed and reducing memory requirements. Beyond these practical constraints, another critical concern is the risk of adversarial attacks (Chakraborty et al., 2018). These attacks craft perturbations to input data that – while appearing benign or imperceptible to humans – can mislead a machine learning model into making incorrect predictions or classifications. The potential implications of successful adversarial attacks are considerable, particularly in critical applications such as autonomous driving, smart health, and fraud detection (Eykholt et al., 2018). The increasing reliance on machine learning models in mission-critical IoT and edge devices further underscores the importance of studying the relationship between model compression and adversarial robustness.

Previous work has largely focused on pruning models that have already undergone adversarial training and have demonstrated robustness (Ye et al., 2019; Cheng et al., 2017; Jordao & Pedrini, 2021). Their primary objective is to investigate how to compress the model without nullifying the effects of adversarial training or undermining the methods that have been implemented to enhance adversarial robustness. However, there are cases where prior adversarial training may not be feasible – there is often a large computation cost of robust/adversarial training (Wang et al., 2020). Furthermore, transferable adversarial samples have been shown to overcome adversarial training (Tramèr et al., 2017). Therefore, it becomes important to understand the impact of adversarial attacks under scnearios where robustness is not already guaranteed. Other works have also shown the effect pruning and compression can have on improving model generalization (Jin et al., 2022). Our focus is on understanding the effects of pruning dense models that have not been adversarially trained or that have not undergone any adversarial robustness enhancement.

In this paper, we establish a comprehensive benchmark to evaluate adversarial robustness in pruned convolution neural networks (CNNs). Our aim is to provide a detailed understanding of the effects of existing pruning methods op-

[*]Equal contribution [1]Department of Computer Science, University of California, Davis [2]Department of Electrical and Computer Engineering, University of California, Davis. Correspondence to: Brijesh Vora <bhvora@ucdavis.edu>, Kartik Patwari <kpatwari@ucdavis.edu>.

2nd AdvML Frontiers workshop at 40th International Conference on Machine Learning, Honolulu, Hawaii, USA. PMLR 202, 2023.
Copyright 2023 by the author(s).

timized for accuracy on models that are not already offering adversarial robustness. We consider a range of adversarial attacks (more details in Section 3.3).

We generate adversarial inputs using multiple attacks on the full and dense models, hereafter referred to as 'base' models. Subsequently, we evaluate the effectiveness of these adversarial inputs on various pruned versions of the respective base models. We consider this a realistic threat model, given that large and densely trained models such as ResNets (He et al., 2016) are widely and publicly available in terms of both architecture and weights. These can be utilized by an attacker as surrogate models for the adversarial example crafting procedure.

Our results reveal that the pruning process has a negligible impact on the adversarial accuracy of the models. More specifically, the adversarial robustness of these models neither significantly deteriorates nor improves post-pruning while providing the benefits of pruning – increased inference speed and better generalizability. We further extend our investigations to explore the transferability of adversarial examples across different model architectures/families. In these experiments, adversarial examples are generated from a base model architecture or family and are then fed to other base models and their pruned counterparts. These tests exhibit the same pattern as earlier, reinforcing our findings – the adversarial impact on pruned models aligns closely with that of their base models.

2. Background and Related Works

2.1. Adversarial Attacks

Adversarial (evasion) attacks aim to craft input samples that cause misclassification by models while appearing visually similar to the original input. Adversarial attacks have continued to evolve, starting with the Fast Gradient Sign Method (FGSM) (Goodfellow et al., 2014) and advancing to multi-step iterative methods like Projected Gradient Descent (PGD) (Madry et al., 2017) and optimization-based attacks like Carlini Wagner (CW) (Carlini & Wagner, 2017). Attacks have been rapidly growing since, as well as reflecting a dynamic cycle: as attacks grow more sophisticated, defenses adapt in response, driving swift progress in the field on both attack and defense fronts. Recent works have introduced more complex iterative methods and optimization-based attacks (Dong et al., 2018; Croce & Hein, 2020; Xu, 2020; Wang et al., 2021; Chen et al., 2018; Wong et al., 2019; Ghiasi et al., 2020). Novel attacks such as adversarial patch attacks (Liu et al., 2018) and adversarial examples in the physical world (Dong et al., 2022) have also emerged. Advancements have also been seen in enhancing attack robustness through adversarial transferability (Guo et al., 2019; Chen et al., 2020b; Andriushchenko et al., 2020) and utilization of Generative Adversarial Networks (GANs) for attack generation (Xiao et al., 2018; Mao et al., 2020).

2.2. Neural Network Pruning

The goal of network pruning is to eliminate redundant or unimportant connections and parameters from a neural network while maintaining or improving its performance, with techniques applied before, after, or even during training.

Post-training pruning techniques remove connections or filters based on their magnitude or contribution to the output, applied either once (single-shot) or iteratively (Han et al., 2016; Liu et al., 2017; He et al., 2017; Yu et al., 2019). One-shot pruning aims to remove a large portion of the network in a single step (Molchanov et al., 2017; Liu et al., 2019) whereas iterative pruning involves pruning a small portion of the network at a time, then retraining the remaining part of the network (Tan & Motani, 2020; Chijiwa et al., 2021; Han et al., 2015).

Pre-training pruning is a technique applied before training the model where the objective is to initialize a smaller network that can be trained from scratch. The lottery ticket hypothesis (Frankle & Carbin, 2019a) introduced the concept of "winning tickets" in neural networks, which are subnetworks that can be trained in isolation to achieve comparable performance to the original network. Since then, various works have built upon this idea (Frankle & Carbin, 2019b; Evci et al., 2022; Frankle et al., 2020)

2.3. Pruning Adversarially Robust Networks

There has been a growing interest in pruning techniques that sustain the robustness of adversarially trained neural networks. Ye et al. (Ye et al., 2019) proposed a joint loss function comprising the compression rate and a robustness term, which guided the pruning of weights with the lowest L_1 norm. Sehwag et al. (Sehwag et al., 2020) introduced a strategy that jointly optimizes the network's accuracy and adversarial robustness during pruning, achieved by adding a robustness-encouraging regularization term. Bai et al. (Bai et al., 2021) developed a Channel-wise Activation Suppressing (CAS) strategy to enhance a network's adversarial robustness by suppressing redundant activation based on their observation of uniform channel activation by adversarial samples. Lim et al. (Lim et al., 2021) presented a robustness-aware filter pruning algorithm that prunes convolution layer filters based on their robustness contribution, calculated by the network's output sensitivity to each filter's removal. Lastly, Li et al. (Li et al., 2022) proposed a pruning algorithm focused on neuron instability as an adversarial perturbation sensitivity metric, removing the most unstable neurons to maintain robustness.

Investigating the interplay between pruning and adversar-

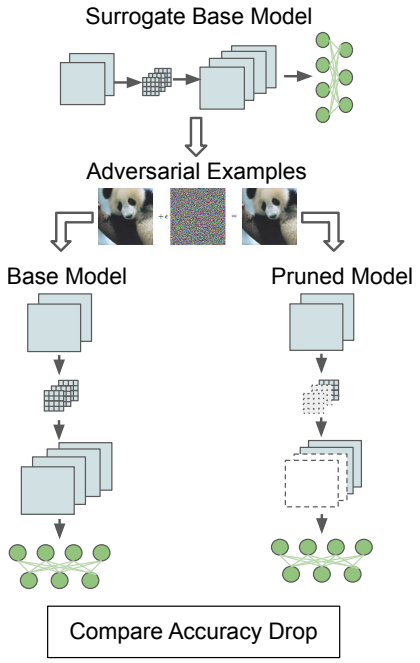

Figure 1. Benchmark pipeline. Adversarial examples generated by various attacks from (attacker's) surrogate base models and evaluated on (victim's) base and pruned model.

ial robustness is crucial. This exploration extends beyond adversarially trained base models, which may not always be feasible or available due to large computation demands (Zhang et al., 2019; Madry et al., 2017) and difficulty of mining samples for robust training (Shrivastava et al., 2016). Hence, we explore how adversarial robustness is affected while pruning conventionally trained models. The benchmarks we present are unique and unexplored in the literature.

3. Methodology

3.1. Threat Model

Our benchmark pipeline, which is indicative of our threat model, is illustrated in Figure 1.

Attacker's Goal: In adversarial attacks, the attacker can have multiple goals, being an untargeted or targeted attack. In a targeted attack, the attacker manipulates an input to make the victim model predict a specifically chosen incorrect class. Conversely, an untargeted adversarial attack aims to induce any incorrect classification without targeting a specific wrong class. Similar to the prior recent adversarial robustness benchmark (on vision transformers) by Mahmood et al. (Mahmood et al., 2021), we consider the untargeted attack scenario.

Attacker's Knowledge: We consider a white-box adversary model, which is often chosen for benchmarking adversarial attacks (Mahmood et al., 2021; Dong et al., 2020). In the white-box attack scenario, the adversary typically has full knowledge of the victim model's architecture and parameters. However, in our relaxed threat model, we assume the adversary lacks knowledge of the victim's trained base model parameters. Instead, the adversary can train a surrogate base model on the same dataset to generate adversarial examples.

Attacker's Capability: The attacker can manipulate inputs by adding perturbations (noise), which are small enough to be imperceptible by human inspection. All attacks we analyze typically confine adversarial perturbations within the bounds of the l_p norm, which is standard. Details about further attack parameters, such as epsilon and iteration values, are provided under Appendix A.1.

3.2. Base Models

For our base models, we choose well-known and readily available models, including ResNet50 (RN50) (He et al., 2016), DenseNet121 (DN121) (Huang et al., 2017), VGG19 (VGG19) (Simonyan & Zisserman, 2014), and MobileNetV1 (NM) (Howard et al., 2017). While the VGGs, ResNets, and DenseNets models are large with millions of parameters, MobileNetV1 has been designed to be lightweight and less dense, providing an interesting point of comparison. Our choice of CNNs from the time and resources in our arsenal. Also, it is worth noting that the time it takes to generate adversarial examples is a few hours to days(specification mentioned in Section 4.1. Therefore, we test the adversarial robustness of the pruned versions of these base models under the assumption that the adversary has access to the base model to craft inputs. We train these models on two standard datasets – CIFAR-10 and CIFAR-100 (Krizhevsky et al., 2009).

3.3. Adversarial Attacks

We consider a subset of popular adversarial attacks that have previously been used to benchmark adversarial robustness (Mahmood et al., 2021; Dong et al., 2020). Hence, our benchmark includes FGSM (Goodfellow et al., 2014), DeepFool (DF) (Moosavi-Dezfooli et al., 2016), PGD (Madry et al., 2017), Basic Iterative Method (BIM) (Kurakin et al., 2018), Auto Projected Gradient Descent (APGD) (Croce & Hein, 2020), and CW (Carlini & Wagner, 2017). We also include Universal Perturbation (UP) attack (Moosavi-Dezfooli et al., 2017), which is a popular attack not considered in the two prior benchmarks. We utilize the Adversarial Robustness Toolbox (ART) library to run the attacks and keep default parameters for each attack provided by ART (see Appendix A.1 for details).

We generate a unique adversarial test set for each attack

using base models – that is, we transform the benign CIFAR-10/CIFAR-100 test set into adversarial samples for every distinct attack. We evaluate robustness by comparing the model's accuracy on the benign test set and each adversarial test set, observing the change in performance.

Table 1. Base & pruned model accuracy on benign CIFAR-10 test set. The pruning target ranges from 10% - 50% with L_1, L_2, and Geometric Median (GM) criterion. Green marked numbers represent the highest achieved accuracy for the specified base model across all pruning specifications, while Red represents the lowest. Here, max δ = max pruned accuracy − base accuracy and min δ = min pruned accuracy − base accuracy.

Pruning Type	Pruning %	Benign Test Accuracy MN	DN121	RN50	VGG19
Base	0%	79.2	83.3	78.1	79.6
L1	10%	**84.4**	**87**	82.1	82.2
L1	20%	82.7	85.5	**82.2**	81.9
L1	30%	82.7	84.6	78.9	81.7
L1	40%	81.4	84.3	80.8	81.6
L1	50%	80.2	82	78.1	**79.7**
L2	10%	83.7	86.9	80.6	82.1
L2	20%	83.5	86.2	79	81.6
L2	30%	81.8	86.7	82	81.3
L2	40%	81.1	85.2	74.9	81.1
L2	50%	**80.0**	81.3	**70.4**	82.1
GM	10%	83.7	86.6	78.6	81.6
GM	20%	82.5	86.2	81.5	81.5
GM	30%	81.4	85.8	83.1	**82.5**
GM	40%	81.2	84.4	79.2	80.3
GM	50%	80.3	**80.1**	78.6	80.1
max δ		**5.2**	**3.7**	**5**	**2.9**
min δ		**0.8**	-3.2	-7.7	**0.1**

3.4. Pruning with NNCF

Our study concentrates on the popular and effective scheme of Iterative Magnitude Pruning (IMP) (Zullich et al., 2021). In IMP, weights beneath a specified magnitude threshold, determined by a pruning criterion, are pruned. This threshold can be set by a predefined sparsity level or a specific percentage of weights with the lowest magnitudes. We use the Neural Network Compression Framework (NNCF) library (Kozlov et al., 2021) to prune base models, employing its filter pruning algorithm. This algorithm iteratively identifies and removes output filters in convolutional layers with the lowest importance, based on filter importance criteria of L_1, L_2, and geometric median. Pruning targets range from 10% to 50% in our study (in 10% increments), meaning up to half of the least important filters are eliminated from the network after the pruning process. Each pruning step is followed by a fine-tuning phase to optimize performance. Details about parameters for fine-tuning can be found in Appendix A.2.

Table 2. Base and pruned model on CIFAR-10. Adversarial Test Accuracies for the base model and the **L2 filter-pruned model** with 10-50% pruning are shown. Examples are generated from the base model and fed to the base and pruned models. Bold numbers represent *positive* maximum δ and the respective maximum value in the pruned model.

Attack	Base	L2-Pruned 10%	20%	30%	40%	50%	max δ
MobileNet							
CW	67.5	65.8	64.9	65.2	63.6	65.2	-1.7
DF	40.7	**43**	**43**	42.3	41.8	41.8	**2.3**
FGSM	12	12.4	12.3	11.3	**12.8**	11.1	**0.8**
BiM	5.1	4.1	3.5	3.8	3.9	**5.6**	**0.5**
PGD	7.4	4.1	4.6	4.2	4.7	6.7	-0.7
APGD	8.5	3.7	4.1	4.4	5.8	8	-0.5
UP	58.8	**64.9**	63	60.1	60.3	59	**6.1**
DenseNet121							
CW	72.1	70	68.1	69.9	69.1	66.9	-2.1
DF	38.7	38.1	38.2	**38.8**	37.8	35.6	**0.1**
FGSM	11.3	11.6	**12**	10.9	10.6	10.3	**0.7**
BiM	8	6.5	6.1	6.7	6.5	6.7	-1.3
PGD	7.1	6.7	6.8	6.8	6.7	7.1	0
APGD	4.5	3.9	4	3.8	3.6	3.8	-0.5
UP	10.2	10	10.6	**11.7**	10.3	10	**1.5**
ResNet50							
CW	68.3	71.4	69.4	**72.7**	65	60.4	**4.4**
DF	37.1	39.7	39.7	**40.4**	35.9	33.8	**3.3**
FGSM	15.4	11.9	13.6	13.6	13.2	13	-1.8
BiM	7	7.2	**7.6**	7.4	7.4	6.9	**0.6**
PGD	7.8	7.5	7.5	7.5	7.1	**8.3**	**0.5**
APGD	4.9	3.9	4.2	4.1	4.4	**6.5**	**1.6**
UP	15.6	14.1	11.8	14.9	11.4	11	-0.7
VGG19							
CW	67.2	**70.2**	69.6	70	68.7	70.1	**3**
DF	38.9	**39**	38.8	**39**	37.1	38.1	**0.1**
FGSM	17.8	16.9	17.6	17.4	**18.1**	16.9	**0.3**
BiM	9.2	9.2	9.2	9.2	9.1	**9.3**	**0.1**
PGD	7.7	7.5	7.8	7.3	**8.3**	7.6	**0.6**
APGD	4.4	3.8	3.8	4	4	4.1	-0.3
UP	49.4	48.7	48.9	**50.6**	47.3	48.8	**1.2**

4. Results

4.1. Filter Pruning

Table 1 shows the benign test accuracies with the base model and pruned models. We observe that the benign test accuracy for all the model families initially increases (with pruning 10%-40%), offering better generalizability and model performance while reducing the number of parameters. We notice this improvement is in part due to the fine-tuned training after every pruning step. At about 50% pruning rate, we see that accuracy is generally similar to the original base model. Notably, there is one outlier in this scenario, ResNet50 – with L2 pruning up to 50%, which has a 7.7% drop-off. **Overall, we find that NNCF's iterative magnitude filter**

Table 3. Inference time (ms) of base and L_2-pruned models on CIFAR-10. Inference time is calculated averaged over 100 runs with batch size = 64.

Model	Inference Time (ms)			
	Base	10%	30%	50%
VGG19	113.80 ± 7.81	104.95 ± 8.81	101.13 ± 6.12	103.65 ± 5.56
RN50	95.07 ± 4.82	88.35 ± 4.29	89.40 ± 5.71	81.41 ± 3.82
MN	23.48 ± 2.46	20.86 ± 2.08	20.39 ± 1.41	21.46 ± 1.97
DN121	71.13 ± 5.27	65.10 ± 3.80	61.09 ± 4.17	60.55 ± 4.05

pruning results in compressed models with fewer parameters and better generalizability, and fine-tuning helps achieve close to the original base model performance, if not better. Benchmarks on CIFAR-100 – presented in Appendix A.4 – demonstrate similar trends.

Table 3 shows the inference time of base and various L2-pruned models. The inference results were generated on a CPU (AMD EPYC 7302 16-Core Processor @ 1.49GHz, 256GB of RAM). We aimed to mimic the constraints of real-world deployment scenarios where high-end server-class GPUs may not be available and their inference time performance does not truly reflect the operational conditions of constrained devices. Both the base and pruned models, initially in the .h5 format, were converted to the OpenVINO format (NNCF's preferred format) for the purpose of inferencing. We ran inference 100 times using a batch size of 64 and reported both the mean and standard deviation of the results. As expected, we find that as the models are pruned from 0% to 50%, the inference time decreases.

Table 4 shows various sizes of the models after pruning. The pruned model sizes are the same for 10% to 50% for NNCF filter pruning (when measuring .h5 file size). From both these tables, we conclude that the size of the models reduces by around 66% and gets a boost in the inference of about 8-10%. **Thus, we see that filter pruning also helps reduce the model size and inference time.**

Table 4. Base and pruned model sizes (MB) on CIFAR-10.

Model Type	Size (MB)			
	MN	DN121	RN50	VGG19
Base	38	82	271	230
Pruned	13.2	29.4	95	77

4.2. Filter Pruning & Adversarial Robustness

Our primary goal is to understand the effect of iterative filter pruning on adversarial robustness. Table 2 presents the accuracy achieved on the CIFAR-10 adversarial test sets crafted on the base models, comparing it to the performance when the same test set is fed to their respective L2 pruned models. We use the weights of the base models to fine-tune the pruned model after removing 10% of the filters. CIFAR-100 results are shown in Appendix A.4.

Table 5. Base and Pruned model on CIFAR-10. Adversarial Test Accuracy and **GM filter pruned model** with 10- 50% pruning and their Adversarial Test Accuracy are shown. Examples are generated from the base model and fed to the base and pruned models.

Attack	Base	GM Pruned					max
		10%	20%	30%	40%	50%	δ
MobileNet							
CW	67.5	64.6	66.5	64.7	64.3	67.5	0
DF	40.7	**44**	42.4	40.1	42.2	41.8	**3.3**
FGSM	12	**21.8**	12.5	13.6	11.9	12.1	**9.8**
BiM	5.1	**8.4**	3.5	3.8	3.8	4.4	**3.3**
PGD	7.4	**18.3**	4.6	4.6	5	5.8	**10.9**
APGD	8.5	**19.3**	4.9	5	5.1	5.6	**10.8**
UP	58.8	**66.9**	63	60.4	57.8	57	**8.1**
DenseNet121							
CW	72.1	58.5	69.8	69.5	68.3	65.1	-2.3
DF	38.7	35.7	37.9	36.6	35.7	36	-0.8
FGSM	11.3	**24.6**	11.6	11.8	11.3	9.5	**13.3**
BiM	8	**8.5**	6.1	8	6.4	6.7	**0.5**
PGD	7.1	**18.4**	6.8	6.7	6.7	6.6	**11.3**
APGD	4.5	**18.6**	3.9	3.9	3.7	3.7	**14.1**
UP	10.2	**16.3**	10.7	10.4	9.7	10.1	**6.1**
ResNet50							
CW	68.3	67.4	71.5	**73.6**	68.9	67.7	**5.3**
DF	37.1	34.8	37.5	**40.6**	38.5	36.7	**3.5**
FGSM	15.4	13.7	11.5	14.3	12.9	11.4	-1.1
BiM	7	7.2	**7.3**	7.3	7.3	7.2	**0.3**
PGD	7.8	7.2	7.4	7.5	7.3	7.5	-0.3
APGD	4.9	4	4	4.4	4.4	4.4	-0.5
UP	15.6	11.7	11.5	10.8	11.9	11.3	-3.7
VGG19							
CW	67.2	**70.9**	69.6	70.4	69.4	66.7	**3.7**
DF	38.9	38.7	**39**	38.9	36.3	37	**0.1**
FGSM	17.8	15.6	18.3	**18.4**	17.2	17	**0.6**
BiM	9.2	9.2	**9.3**	9.3	9.2	9.2	**0.1**
PGD	7.7	7.7	7.8	7.5	8.4	**8.6**	**0.9**
APGD	4.4	3.9	3.7	3.8	3.8	4.2	-0.2
UP	49.4	49.7	51.3	**50.1**	45.3	48.3	**1.9**

Our findings indicate a minimal impact on adversarial accuracy stemming from the pruning process, thus suggesting a relative consistency in the adversarial robustness of the models. Specifically, our experimental results highlight that the process of pruning neither significantly degrades nor enhances the robustness of the models when exposed to adversarial attacks. Primarily the maximum change in accuracy (max δ) is between $\pm 1\%$, with some attacks being slightly higher. The most significant change occurs using UP attack on MobileNet, where the 10% pruned MobileNet's accuracy on UP adversarial test set increases from its base model by about 6%. Our findings underline an intriguing invariance in adversarial robustness when examples are generated from a base model and fed into its pruned models with no prior adversarial robustness measures taken. The results for GM pruned models are shown in table 5 and L1 in Appendix A.3. While the same trend applies, we notice for GM pruned models at 10% we see a slight boost in accuracy (3%-10%

increase) for MobileNet and DenseNet121. However, this is not a consistent trend across other models, and as the pruning target increases up to 50%. The NNCF-based pruning models initially removes the 10% unimportant filters and then for a target pruning of 20-50% it iteratively fine tunes and removes the filters and weights, which leads to the PGD and APGD attack to show significant accuracy change. Furthermore, the accuracy of the CW is $>= 60\%$ because of the adversarial examples are generated using the surrogate base model using a similar training scheme. **Overall, our results show that pruning base models result in compressed models that run faster while maintaining comparable performance and adversarial robustness.**

Table 6. Transferability results on CIFAR-10. The adversarial test sets are generated from (surrogate) ResNet50 and fed to the victim models with different architectures. Accuracies shown are for victim models. Bold numbers represent *positive* maximum δ and the respective maximum value in the pruned model.

Pruning %	Surrogate Model: ResNet50						
L2	APGD	BiM	CW	DF	FGSM	PGD	UP
MobileNet							
0%	13.5	9.3	78.5	41	12.6	12.5	11.7
10%	9.1	9.1	**83**	**45**	**13**	9	11.3
20%	9.2	9.2	82.3	44	12.5	9.6	11.2
30%	8.7	9	81.6	43	12.8	8.5	11.2
40%	11.1	9.7	79.2	41.4	12.1	10.3	**12**
50%	10.8	**9.9**	78.1	41	10.9	9.8	11.4
max δ	-2.4	**0.6**	**4.5**	**4**	**0.4**	-2.2	**0.3**
DenseNet121							
0%	10.4	11.7	83	43.3	14.3	11.5	11.6
10%	6.5	8.7	**85.7**	**45.3**	12.4	8.2	10
20%	6.8	7.4	84.6	44.3	12.4	8.1	10.1
30%	6.2	8.2	85.3	44.4	12.2	8.6	9.9
40%	5.8	7.4	84.8	43.9	12.4	8.6	11
50%	6.1	7.2	80.5	39.6	12.1	9.7	10
max δ	-3.6	-3	**2.7**	**2**	-1.9	-1.8	-0.6
VGG19							
0%	7.9	8.5	78.5	41.3	17.2	8.6	12.4
10%	9.1	8.5	**81.5**	43.5	15.4	8.2	10.9
20%	**9.5**	7.6	80.5	43.7	15.5	8.5	12
30%	7.3	7.6	80.2	41.1	15.4	8.2	12.6
40%	8.6	8.1	80.5	43.5	15.6	8.5	**14.5**
50%	7.1	8.2	81.1	**44.6**	14.4	8	10.1
max δ	**1.6**	0	**3**	**3.3**	-1.6	-0.1	**2.1**

4.3. Adversarial Transferability

This study explores the effect of feeding adversarial examples generated from one model architecture family into different model families. This departs from the previous sections, where adversarial examples were tested within the same model family. Here, we consider adversarial test sets or examples created from the surrogate base model – ResNet50, and these are now cross fed into all other base models and their respective pruned models. The focus of this investigation is to understand the phenomena and implications of adversarial transferability (Guo et al., 2019; Chen et al., 2020b; Andriushchenko et al., 2020) across different pruned architectures. Table 6 demonstrates our findings when ResNet50 is used as the surrogate base model for adversarial example generation. Results for the remaining models as surrogate models demonstrate similar trends (see Appendix A.3.1). We observe that the pruned models do not show any significant variations in their adversarial transferability compared to their base models. This corroborates our primary findings from Section 4.2. **Even in a cross-model testing environment, the pruned models exhibit adversarial robustness comparable to their original base models.**

5. Conclusion and Future Work

In conclusion, this paper presents a rigorous evaluation of the impact of filter pruning on the adversarial robustness of neural network models. We demonstrate, using the CIFAR-10 and CIFAR-100 datasets, that despite compressing the model by up to 50% through filter pruning, the adversarial robustness remains relatively unaffected as compared to the base models. Our results offer promising implications, indicating that while practitioners can enjoy the benefits of filter pruning – such as accelerated inference time, curtailment in over-parameterization, and enhanced generalization capabilities – they do not have to compromise on adversarial resilience. Finally, our study supports the application of filter pruning, showcasing no detrimental effects on adversarial robustness with respect to the original base model.

One crucial area for future investigation involves the exploration of alternative compression techniques that go beyond the scope of NNCF-based pruning. It is essential to expand the repertoire of compression methods to find innovative approaches that can further enhance the efficiency of neural network models. By venturing into unexplored territories, researchers can discover novel ways to compress models effectively, reducing their size and computational requirements while maintaining high performance, including comparable robustness against evasion attacks. Furthermore, we encourage future research to continue exploring the intersection of model compression and adversarial robustness, contributing further to the creation of efficient, secure, and robust models ready for real-world deployment.

Acknowledgements

We thank the anonymous reviewers for their helpful comments. This project is based upon work supported in part by the UC Noyce Institute: Center for Cybersecurity and Cyberintegrity (C-CUBE).

References

Andriushchenko, M., Croce, F., Flammarion, N., and Hein, M. Square attack: a query-efficient black-box adversarial attack via random search. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XXIII*, pp. 484–501. Springer, 2020.

Bai, Y., Zeng, Y., Jiang, Y., Xia, S.-T., Ma, X., and Wang, Y. Improving adversarial robustness via channel-wise activation suppressing. *arXiv preprint arXiv:2103.08307*, 2021.

Carlini, N. and Wagner, D. Towards evaluating the robustness of neural networks. In *2017 ieee symposium on security and privacy (sp)*, pp. 39–57. Ieee, 2017.

Chakraborty, A., Alam, M., Dey, V., Chattopadhyay, A., and Mukhopadhyay, D. Adversarial attacks and defences: A survey. *arXiv preprint arXiv:1810.00069*, 2018.

Chen, C., Zhang, P., Zhang, H., Dai, J., Yi, Y., Zhang, H., and Zhang, Y. Deep learning on computational-resource-limited platforms: a survey. *Mobile Information Systems*, 2020:1–19, 2020a.

Chen, J., Jordan, M. I., and Wainwright, M. J. Hopskipjumpattack: A query-efficient decision-based attack. In *2020 ieee symposium on security and privacy (sp)*, pp. 1277–1294. IEEE, 2020b.

Chen, P.-Y., Sharma, Y., Zhang, H., Yi, J., and Hsieh, C.-J. Ead: elastic-net attacks to deep neural networks via adversarial examples. In *Proceedings of the AAAI conference on artificial intelligence*, volume 32, 2018.

Cheng, Y., Wang, D., Zhou, P., and Zhang, T. A survey of model compression and acceleration for deep neural networks. *arXiv preprint arXiv:1710.09282*, 2017.

Chijiwa, D., Yamaguchi, S., Ida, Y., Umakoshi, K., and Inoue, T. Pruning randomly initialized neural networks with iterative randomization. *Advances in Neural Information Processing Systems*, 34:4503–4513, 2021.

Croce, F. and Hein, M. Reliable evaluation of adversarial robustness with an ensemble of diverse parameter-free attacks. In *International conference on machine learning*, pp. 2206–2216. PMLR, 2020.

Dong, Y., Liao, F., Pang, T., Su, H., Zhu, J., Hu, X., and Li, J. Boosting adversarial attacks with momentum. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 9185–9193, 2018.

Dong, Y., Fu, Q.-A., Yang, X., Pang, T., Su, H., Xiao, Z., and Zhu, J. Benchmarking adversarial robustness on image classification. In *proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 321–331, 2020.

Dong, Y., Zhu, J., Gao, X.-S., et al. Isometric 3d adversarial examples in the physical world. *Advances in Neural Information Processing Systems*, 35:19716–19731, 2022.

Evci, U., Ioannou, Y., Keskin, C., and Dauphin, Y. Gradient flow in sparse neural networks and how lottery tickets win. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 36, pp. 6577–6586, 2022.

Eykholt, K., Evtimov, I., Fernandes, E., Li, B., Rahmati, A., Xiao, C., Prakash, A., Kohno, T., and Song, D. Robust physical-world attacks on deep learning visual classification. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 1625–1634, 2018.

Frankle, J. and Carbin, M. The lottery ticket hypothesis: Finding sparse, trainable neural networks. In *International Conference on Learning Representations*, 2019a.

Frankle, J. and Carbin, M. Stabilizing the lottery ticket hypothesis. *arXiv preprint arXiv:1903.01611*, 2019b.

Frankle, J., Dziugaite, G. K., Roy, D., and Carbin, M. Linear mode connectivity and the lottery ticket hypothesis. In *International Conference on Machine Learning*, pp. 3259–3269. PMLR, 2020.

Ghiasi, A., Shafahi, A., and Goldstein, T. Breaking certified defenses: Semantic adversarial examples with spoofed robustness certificates. *arXiv preprint arXiv:2003.08937*, 2020.

Goodfellow, I. J., Shlens, J., and Szegedy, C. Explaining and harnessing adversarial examples. *arXiv preprint arXiv:1412.6572*, 2014.

Guo, C., Gardner, J., You, Y., Wilson, A. G., and Weinberger, K. Simple black-box adversarial attacks. In *International Conference on Machine Learning*, pp. 2484–2493. PMLR, 2019.

Han, S., Pool, J., Tran, J., and Dally, W. Learning both weights and connections for efficient neural network. *Advances in neural information processing systems*, 28, 2015.

Han, S., Mao, H., and Dally, W. J. Deep compression: Compressing deep neural networks with pruning, trained quantization and huffman coding. In *International Conference on Learning Representations*, 2016.

He, K., Zhang, X., Ren, S., and Sun, J. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 770–778, 2016.

He, Y., Zhang, X., and Sun, J. Channel pruning for accelerating very deep neural networks. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 41(1):145–157, 2017.

Howard, A. G., Zhu, M., Chen, B., Kalenichenko, D., Wang, W., Weyand, T., Andreetto, M., and Adam, H. Mobilenets: Efficient convolutional neural networks for mobile vision applications. *arXiv preprint arXiv:1704.04861*, 2017.

Huang, G., Liu, Z., Van Der Maaten, L., and Weinberger, K. Q. Densely connected convolutional networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 4700–4708, 2017.

Jin, T., Carbin, M., Roy, D. M., Frankle, J., and Dziugaite, G. K. Pruning's effect on generalization through the lens of training and regularization. *arXiv preprint arXiv:2210.13738*, 2022.

Jordao, A. and Pedrini, H. On the effect of pruning on adversarial robustness. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 1–11, 2021.

Kozlov, A., Lazarevich, I., Shamporov, V., Lyalyushkin, N., and Gorbachev, Y. Neural network compression framework for fast model inference. In *Intelligent Computing: Proceedings of the 2021 Computing Conference, Volume 3*, pp. 213–232. Springer, 2021.

Krizhevsky, A., Hinton, G., et al. Learning multiple layers of features from tiny images. 2009.

Kurakin, A., Goodfellow, I. J., and Bengio, S. Adversarial examples in the physical world. In *Artificial intelligence safety and security*, pp. 99–112. Chapman and Hall/CRC, 2018.

Li, Z., Chen, T., Li, L., Li, B., and Wang, Z. Can pruning improve certified robustness of neural networks? *arXiv preprint arXiv:2206.07311*, 2022.

Lim, H., Roh, S.-D., Park, S., and Chung, K.-S. Robustness-aware filter pruning for robust neural networks against adversarial attacks. In *2021 IEEE 31st International Workshop on Machine Learning for Signal Processing (MLSP)*, pp. 1–6. IEEE, 2021.

Liu, X., Yang, H., Liu, Z., Song, L., Li, H., and Chen, Y. Dpatch: An adversarial patch attack on object detectors. *arXiv preprint arXiv:1806.02299*, 2018.

Liu, Z., Li, J., Shen, Z., Huang, G., Yan, S., and Zhang, C. Learning efficient convolutional networks through network slimming. In *Proceedings of the IEEE International Conference on Computer Vision*, pp. 2736–2744, 2017.

Liu, Z., Sun, M., Zhou, T., Huang, G., and Darrell, T. Rethinking the value of network pruning. In *Proceedings of the International Conference on Computer Vision*, pp. 4978–4986, 2019.

Madry, A., Makelov, A., Schmidt, L., Tsipras, D., and Vladu, A. Towards deep learning models resistant to adversarial attacks. *arXiv preprint arXiv:1706.06083*, 2017.

Mahmood, K., Mahmood, R., and Van Dijk, M. On the robustness of vision transformers to adversarial examples. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 7838–7847, 2021.

Mao, X., Chen, Y., Li, Y., He, Y., and Xue, H. Gap++: Learning to generate target-conditioned adversarial examples. *arXiv preprint arXiv:2006.05097*, 2020.

Molchanov, D., Ashukha, A., and Vetrov, D. Variational dropout sparsifies deep neural networks. *Journal of Machine Learning Research*, 18(1):2118–2136, 2017.

Moosavi-Dezfooli, S.-M., Fawzi, A., and Frossard, P. Deepfool: a simple and accurate method to fool deep neural networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 2574–2582, 2016.

Moosavi-Dezfooli, S.-M., Fawzi, A., Fawzi, O., and Frossard, P. Universal adversarial perturbations. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 1765–1773, 2017.

Sehwag, V., Wang, S., Mittal, P., and Jana, S. Hydra: Pruning adversarially robust neural networks. *Advances in Neural Information Processing Systems*, 33:19655–19666, 2020.

Shrivastava, A., Gupta, A., and Girshick, R. Training region-based object detectors with online hard example mining. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 761–769, 2016.

Simonyan, K. and Zisserman, A. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014.

Tan, C. M. J. and Motani, M. Dropnet: Reducing neural network complexity via iterative pruning. In *International Conference on Machine Learning*, pp. 9356–9366. PMLR, 2020.

Tramèr, F., Papernot, N., Goodfellow, I., Boneh, D., and McDaniel, P. The space of transferable adversarial examples. *arXiv preprint arXiv:1704.03453*, 2017.

Wang, X., Lin, J., Hu, H., Wang, J., and He, K. Boosting adversarial transferability through enhanced momentum. *arXiv preprint arXiv:2103.10609*, 2021.

Wang, Y., Zou, D., Yi, J., Bailey, J., Ma, X., and Gu, Q. Improving adversarial robustness requires revisiting misclassified examples. In *International Conference on Learning Representations*, 2020.

Wong, E., Schmidt, F., and Kolter, Z. Wasserstein adversarial examples via projected sinkhorn iterations. In *International Conference on Machine Learning*, pp. 6808–6817. PMLR, 2019.

Xiao, C., Li, B., Zhu, J.-Y., He, W., Liu, M., and Song, D. Generating adversarial examples with adversarial networks. *arXiv preprint arXiv:1801.02610*, 2018.

Xu, J. Generate adversarial examples by nesterov-momentum iterative fast gradient sign method. In *2020 ieee 11th international conference on software engineering and service science (icsess)*, pp. 244–249. IEEE, 2020.

Ye, S., Xu, K., Liu, S., Cheng, H., Lambrechts, J.-H., Zhang, H., Zhou, A., Ma, K., Wang, Y., and Lin, X. Adversarial robustness vs. model compression, or both? In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 111–120, 2019.

Yu, X., Liu, S., Kumar, S., and Cheng, Y. Nisp: Pruning networks using neuron importance score propagation. In *International Conference on Learning Representations*, 2019.

Zhang, H., Yu, Y., Jiao, J., Xing, E., El Ghaoui, L., and Jordan, M. Theoretically principled trade-off between robustness and accuracy. In *International conference on machine learning*, pp. 7472–7482. PMLR, 2019.

Zhao, C., Ni, B., Zhang, J., Zhao, Q., Zhang, W., and Tian, Q. Variational convolutional neural network pruning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 2780–2789, 2019.

Zullich, M., Medvet, E., Pellegrino, F. A., and Ansuini, A. Speeding-up pruning for artificial neural networks: introducing accelerated iterative magnitude pruning. In *2020 25th International Conference on Pattern Recognition (ICPR)*, pp. 3868–3875. IEEE, 2021.

A. Appendix

A.1. Adversarial Attack Details

Table 7. Attack details

Attack	Epsilon	Epsilon Step	Max iteration
CW	-	0.01	10
DF	1e-6	-	5
FGSM	0.2		100
BiM	1	0.1	100
PGD	0.3	0.1	100
APGD	0.3	0.1	100
UP	10	-	1

A.2. Neural Network Pruning Details

The batch size is 128 and 10 epochs for (base) training. The models have been trained on CIFAR-10/CIFAR-100 trainset. We choose the optimizer as SGD with a learning rate of 0.1, gamma as 0.1 and steps as [10, 20, 30]. The momentum of 0.9 and nesterov is set to true. The schedule_step is multistep. For compression we use algorithm as filter_pruning, and schedule as exponential. The pruning_init is set to 0.1 for all the L_1 and L_2 and GM but the pruning_target varies from 0.1 to 0.5 [10 - 50%]. The filter_importance is [L_1 and L_2 and GM].

A.3. CIFAR-10 Results

Table 8. Base and Pruned model on CIFAR-10. Adversarial Test Accuracy and **L1 filter pruned model** with 10- 50% pruning and their Adversarial Test Accuracy are shown. Examples are generated from the base model and fed to the base and pruned models.

Attack	Base	L1 Pruned					max
		10%	20%	30%	40%	50%	δ
MobileNet							
CW	67.5	65.6	65.8	66.2	66.2	65.3	-1.3
DF	40.7	42.7	**43.3**	41.2	39.4	40.1	**2.6**
FGSM	12	12.3	12.5	**13.7**	13.1	12.8	**1.7**
BiM	5.1	4.6	4.1	3.8	3.1	3.1	-0.5
PGD	7.4	4.8	4.5	4.7	4.6	4.4	-2.6
APGD	8.5	3.8	4	4.4	4.6	4.6	-3.9
UP	58.8	65	60.4	**61.2**	56.7	56.6	**6.2**
DenseNet121							
CW	72.1	70.3	69.1	68.1	67.9	69.7	-1.8
DF	38.7	38	36.2	33.3	34.9	35.7	-0.7
FGSM	11.3	11.9	12	**12.7**	11.4	9.8	**1.4**
BiM	8	7.7	**8.4**	6.2	6.4	7.2	**0.4**
PGD	7.1	6.7	7.1	6.8	6.8	**8**	**0.9**
APGD	4.5	4	4	3.9	3.7	3.7	-0.5
UP	10.2	10	9.8	10.2	10	10	0
ResNet50							
CW	68.3	73.1	**72.4**	69.2	71.2	68.1	**4.8**
DF	37.1	39.8	**40.2**	39.3	38.7	37.8	**3.1**
FGSM	15.4	12.4	12.2	13.1	13.1	11.9	-2.3
BiM	7	7.3	7.6	7.4	**7.7**	7.3	**0.7**
PGD	7.8	7.6	7.5	7.5	7.2	7.5	-0.2
APGD	4.9	4.1	4	3.8	3.7	4.7	-0.2
UP	15.6	11.7	15.2	11.8	11.9	**17.3**	**1.7**
VGG19							
CW	67.2	**70.4**	70.4	69.2	69.6	67.6	**3.2**
DF	38.9	38.8	**39.2**	38.9	37.2	38.8	**0.3**
FGSM	17.8	16.5	18	17.8	**18.8**	15.3	**1**
BiM	9.2	9.2	9.2	**9.3**	9.1	9.4	**0.2**
PGD	7.7	7.6	7.9	7.8	**8.1**	7.7	**0.4**
APGD	4.4	4.1	4	**4.5**	4.3	4.4	**0.1**
UP	49.4	**49.9**	48.1	48.3	47.6	46.6	**0.5**

A.3.1. ADVERSARIAL TRANSFERABILITY

Table 9. Transferability results on CIFAR-10. The adversarial test sets generated from (surrogate) DenseNet121 and fed to the victim models with different architectures. Accuracies shown are for victim models.

Pruning %	Surrogate Model: DenseNet121						
L2	APGD	BiM	CW	DF	FGSM	PGD	UP
MobileNet							
0%	7.1	8.2	78.2	39.2	14	9.6	12.6
10%	4.5	6.8	81.8	42.4	11.7	8.9	11.3
20%	3.8	7	81.5	41.4	13.5	8.9	11.2
30%	4.9	7.8	80.2	39.3	12.2	9.4	11.2
40%	5.5	9.2	80	39	12.4	9.6	11.2
50%	6	9.7	77.8	37	12.1	11.2	11.3
ResNet50							
0%	8.6	9.4	76.2	36.9	18.3	10.6	14.2
10%	3.8	6.6	78.8	40.2	17.8	8.5	13.2
20%	3.7	9.2	77.9	37.9	18.7	9.2	11.6
30%	4.1	5.8	80.8	40.4	17.2	7.8	13.5
40%	5.8	8.6	73.6	38	19.1	12.5	9
50%	7.9	7.4	68.7	34.4	17	11.7	10.6
VGG19							
0%	4.5	8.8	78.2	38.9	18.7	9.1	10.9
10%	4.1	8.4	80.5	41.6	18.2	9.3	10.7
20%	4.1	8.2	80.2	40.5	17.9	10.4	11.2
30%	4.4	8.5	80.5	40.7	15.8	10.5	12.2
40%	5.1	8.8	79.5	39.8	17.5	10.8	12.5
50%	4.2	7.5	81.2	41.2	16.7	3.8	8.4

Table 10. Transferability results on CIFAR-10. The adversarial test sets generated from (surrogate) MobileNet and fed to the victim models with different architectures. Accuracies shown are for victim models.

Pruning %	Surrogate Model: MobileNet						
L2	APGD	BiM	CW	DF	FGSM	PGD	UP
DenseNet121							
0%	13.3	8.6	82.3	46.5	16.1	11.5	65.2
10%	9.8	9.4	85.5	45	14.5	8.1	70.3
20%	10.5	8.5	84.7	44.1	14.5	8.9	70.2
30%	10.4	9.8	85.6	45.1	13.2	8.5	69.5
40%	10.9	6.1	84.2	44.6	13.4	8.8	68.7
50%	9.7	8.4	79.7	40.2	13.8	8.1	67.9
ResNet50							
0%	18.4	9.5	77	41.1	19.7	15.1	62.5
10%	16.9	8	80.3	45.8	19.4	15	67.3
20%	15	7.9	77.7	45.5	19.7	13.6	65.3
30%	14.8	6.8	81.4	45.2	20.1	13.3	66.5
40%	16	5.8	74.2	41	21.5	14.7	61.7
50%	15.5	9.5	69.6	39.8	19.5	15.7	56.8
VGG19							
0%	16.3	5.4	77.7	42.2	18.6	12.9	67.7
10%	14.6	5.9	81.1	44.6	16.6	9.2	68.7
20%	17.2	5.5	80.4	44.7	16.5	11	69.3
30%	15.6	5.6	80.4	43.3	16.8	10.2	69.3
40%	14.2	6	80.3	43.6	16.6	8.8	67.1
50%	13.7	6	80.8	43.9	15.1	10.3	68

Table 11. Transferability results on CIFAR-10. The adversarial test sets generated from (surrogate) VGG19 and fed to the victim models with different architectures. Accuracies shown are for victim models.

Pruning %	Surrogate Model: VGG19						
L2	APGD	BiM	CW	DF	FGSM	PGD	UP
MobileNet							
0%	12.9	11.3	74.5	49.8	15.9	15.3	49.2
10%	11.5	10.9	82.1	53.5	17.4	10.6	53.4
20%	10.5	12.4	81.6	51.4	16.8	9.7	52.5
30%	11.3	12.1	80.9	50.4	17.7	12	51.9
40%	11.5	12.2	78.6	49.4	16.5	10.5	48.5
50%	12.5	13.1	77.5	49.2	15	11.7	47.9
DenseNet121							
0%	14.7	13.1	81.1	53.6	16.5	15.4	52.8
10%	9.5	9.9	85.3	54.3	19.4	10.7	60.6
20%	9.9	10.2	84.3	54.3	19.2	11.4	61.7
30%	10.3	12.2	84.7	51.4	17.4	11.5	60
40%	12	13.2	83.8	52.4	16.7	12.2	61
50%	11.4	12	79.1	48	16	12.9	56.4
ResNet50							
0%	17.1	12	77.1	49.2	20.8	17.2	53.8
10%	14.6	10.7	79.3	51.8	21.5	11.5	59.7
20%	13.7	12.3	77.8	49.3	21.5	12.8	59.7
30%	10	82.2	82.2	56.2	24.2	12.5	59.9
40%	14.4	10.5	74.1	43.6	24.1	11.1	54.2
50%	16.5	9.4	68.9	41.9	23.7	16	50.7

Table 12. Inference time (ms) of base and L2 pruned models on CIFAR-10. Inference time calculated averaged over 100 runs with batch size = 32

Model	Inference Time(ms)			
	Base	10%	30%	50%
VGG19	102.54 ± 8.35	91.30 ± 4.96	94.23 ± 5.25	93.21 ± 5.49
RN50	93.92 ± 5.50	93.82 ± 4.48	90.42 ± 5.17	87.93 ± 3.63
MN	20.43 ± 1.61	19.41 ± 2.09	19.65 ± 1.98	19.60 ± 1.80
DN121	63.77 ± 5.67	59.39 ± 3.64	59.23 ± 3.88	57.26 ± 4.10

A.4. CIFAR-100 Results

Since we are NNCF-based pruning the models for which we initially remove the 10% unimportant filters and then for a target pruning of 20-50% it iteratively fine tunes and removes the filters and weights which are unimportant to reach target pruning, and for 10% of pruning target it just fine-tunes the model, so that leads to the DF attack as shown in table 14, 15, 16, to show significant accuracy change.

Table 13. Base & pruned model accuracy on benign CIFAR-100 test set. The pruning target ranges from 10% - 50% with L_1, L_2, and Geometric Median (GM) pruning criterion.

Pruning Type	Pruning %	Benign Test Accuracy MN	DN121	RN50
Base	0%	50.33	49.97	43.3
L1	10%	**57.35**	62.11	**34.22**
L1	20%	55.79	60.48	46.34
L1	30%	54.58	58.09	50.68
L1	40%	53.05	57.04	50.54
L1	50%	**51.04**	**53.63**	**51.3**
L2	10%	57.24	62.07	39.32
L2	20%	55.41	61.09	43.49
L2	30%	54.42	59.66	39.83
L2	40%	53.01	58.27	44.03
L2	50%	51.4	53.94	45.97
GM	10%	56.95	**62.17**	41.59
GM	20%	55.22	60.9	44.56
GM	30%	54.51	59.25	46
GM	40%	52.97	58.05	47.38
GM	50%	51.44	54.57	42.6
Max Delta		**7.02**	**12.2**	**8**
Min Delta		**0.71**	**3.66**	**-9.08**

Table 14. Base and Pruned model on CIFAR-100. Adversarial Test Accuracy and **L2 filter pruned model** with 10- 50% pruning and their Adversarial Test Accuracy are shown. Examples are generated from the base model and fed to the base and pruned models.

Attack	Base	L2 Pruned 10%	20%	30%	40%	50%	max δ
MobileNet							
DF	20.6	**35.07**	34.13	33.92	33.25	32.99	**14.47**
FGSM	7.9	2.29	2.45	2.19	2.6	2.49	-5.3
BiM	1.5	0.86	0.84	0.77	0.64	0.66	-0.64
PGD	7.7	0.95	0.95	0.94	1.21	1.29	-6.41
APGD	8.6	0.79	0.86	0.77	1.19	1.07	-7.41
UP	25.5	**39.62**	38.96	36.66	35.18	33.67	**14.12**
DenseNet121							
DF	13.49	**35.76**	35.1	34.18	33.64	31.27	**22.27**
FGSM	1.45	1.61	**1.98**	1.54	1.81	1.2	**0.53**
BiM	0.92	0.9	1.01	**1.23**	0.82	0.82	**0.31**
PGD	3.22	5.95	**6.1**	5.16	5.85	4.07	**2.88**
APGD	1.03	1	1	1.05	0.96	**1.11**	**0.08**
UP	3.75	6.59	6.5	5.6	**6.94**	4.21	**3.19**
ResNet50							
DF	22.1	29.55	31.59	29.2	31.24	**32.94**	**10.84**
FGSM	13.2	4.85	2.43	2.94	1.81	2.52	-8.35
BiM	1.7	0.78	0.79	0.75	0.9	0.76	-0.8
PGD	14.7	**14.72**	7.99	8.41	6.08	7	**0.02**
APGD	11.7	3.56	1.31	1.42	1.21	1.31	-8.14
UP	26.5	20.93	16.65	13.98	11.84	13.03	-5.57

Table 15. Base and Pruned model on CIFAR-100. Adversarial Test Accuracy and **L1 filter pruned model** with 10- 50% pruning and their Adversarial Test Accuracy are shown. Examples are generated from the base model and fed to the base and pruned models.

Attack	Base	L1 Pruned					max
		10%	20%	30%	40%	50%	δ
MobileNet							
DF	20.6	**35.2**	34.14	34.07	32.89	32.37	**14.6**
FGSM	7.9	2.11	2.3	2.37	2.05	2.23	-5.53
BiM	1.5	0.86	0.81	0.74	0.58	0.7	-0.64
PGD	7.7	0.74	0.76	1.07	0.99	1.22	-6.48
APGD	8.6	0.83	0.81	0.99	0.99	1.07	-7.53
UP	25.5	**39.46**	37.55	37	35.91	34.64	**13.96**
DenseNet121							
DF	13.49	**35.97**	34.75	33.4	33.3	30.6	**22.48**
FGSM	1.45	1.56	**1.87**	1.82	1.69	1.11	**0.42**
BiM	0.92	**1.25**	0.97	0.99	1.03	1.06	**0.33**
PGD	3.22	5.62	**5.91**	5.2	4.98	3.75	**2.69**
APGD	1.03	1.07	**1.13**	1.2	1	0.95	**0.17**
UP	3.75	5.67	5.26	5.61	**5.83**	3.04	**2.08**
ResNet50							
DF	22.1	25.66	33.56	36.8	36.77	**37.19**	**15.09**
FGSM	13.2	4.51	3.24	1.64	2.46	2.52	-8.69
BiM	1.7	0.63	0.8	0.82	0.77	0.71	-0.88
PGD	14.7	14.17	8.75	7.23	7.62	7.39	-0.53
APGD	11.7	3.71	1.21	1.26	1.24	1.09	-7.99
UP	26.5	19.9	15.66	13.16	14.23	15.02	-6.6

Table 16. Base and Pruned model on CIFAR-100. Adversarial Test Accuracy and **GM filter pruned model** with 10- 50% pruning and their Adversarial Test Accuracy are shown. Examples are generated from the base model and fed to the base and pruned models.

Attack	Base	GM Pruned					max
		10%	20%	30%	40%	50%	δ
MobileNet							
DF	20.6	**34.83**	33.89	33.93	33.74	33.45	**14.23**
FGSM	7.9	2.27	1.94	2.43	2.2	2.39	-5.47
BiM	1.5	0.83	0.84	0.72	0.67	0.68	-0.66
PGD	7.7	0.87	0.88	0.99	1.12	1.2	-6.5
APGD	8.6	0.89	0.81	1.09	1.29	1.1	-7.31
UP	25.5	**39.72**	37.72	36.62	34.61	34.15	**14.22**
DenseNet121							
DF	13.49	**35.89**	34.59	34.01	33.61	32.05	**22.4**
FGSM	1.45	1.65	**1.8**	1.61	1.4	1.51	**0.35**
BiM	0.92	**1.27**	0.94	1.26	1.24	0.82	**0.35**
PGD	3.22	**6.03**	5.66	4.69	5.62	5.33	**2.81**
APGD	1.03	1.06	0.97	0.88	1.16	**1.31**	**0.28**
UP	3.75	6.56	6.11	4.65	**6.81**	6.27	**3.06**
ResNet50							
DF	22.1	30.69	31.42	34.09	**34.5**	30.29	**12.4**
FGSM	13.2	3.3	2.04	3.95	3.29	2.62	-9.25
BiM	1.7	1.03	0.88	0.91	0.83	0.75	-0.67
PGD	14.7	10.18	6.3	12.05	9.15	7.6	-2.65
APGD	11.7	1.93	1.08	1.88	1.17	1.28	-9.77
UP	26.5	21.67	11.35	25.03	17.9	15.67	-1.47

Table 17. CIFAR-100 -Base and Pruned Model sizes.

Model	Size (MB)		
	Base	Pruned	%Reduction
ResNet50	273	92	66.30
MobileNet	39	13	66.67
DenseNet121	84	29	65.47