Fair Compression of Machine Learning Vision Systems

by

Robbie Meyer

A thesis
presented to the University of Waterloo
in fulfillment of the
thesis requirement for the degree of
Master of Applied Science
in
Systems Design Engineering

Waterloo, Ontario, Canada, 2023

Author's Declaration

This thesis consists of material all of which I authored or co-authored: see Statement of Contributions included in the thesis. This is a true copy of the thesis, including any required final revisions, as accepted by my examiners.

I understand that my thesis may be made electronically available to the public.

Statement of Contributions

The following two papers were used in this thesis. For all papers, I was a co-author with major contributions on the design, development, evaluation and writing of the papers' material.

R. Meyer and A. Wong, "From Intention to Action: The Fair AI Toolbox," *Computational Vision and Imaging Systems*, 2022.

This paper is incorporated in Chapter 2.

R. Meyer and A. Wong, "A Fair Loss Function for Network Pruning," *Trustworthy and Socially Responsible Machine Learning, NeurIPS*, 2022.

This paper is incorporated in Chapters 2, 3 and 4.

Abstract

Model pruning is a simple and effective method for compressing neural networks. By identifying and removing the least influential parameters of a model, pruning is able to transform networks into smaller, faster networks with minimal impact to overall performance. However, recent research has shown that while overall performance may not be significantly changed, model pruning can exacerbate existing fairness issues. Subgroups that are underrepresented or complex may experience a greater than average impact from pruning. Machine learning systems that use compressed neural networks may consequently exhibit significant biases that could limit their effectiveness in many real world situations.

To address this issue, we analyze the effect on fairness of pruning a variety of image classification models and propose a novel method for improving the fairness of existing pruning methods. By analyzing the fairness impact of pruning in a variety of situations, we further our understanding of the theoretical fairness impact of pruning could manifest in real-world conditions. By developing a method for improving the fairness of pruning methods, we demonstrate that the fairness impact of pruning can be influenced and enable machine learning practitioners to improve the post-pruning fairness of their models.

Our analysis revealed that the fairness impact of pruning can be observed in many, but not all, image classification systems that utilize deep learning and pruning. The dataset used to train each model appears to influence how pruning affects the fairness of each model. Models trained and pruned using the CelebA dataset did see a negative impact on fairness while models trained and pruned using the Fitzpatrick17k dataset did not. Manipulating the CelebA and CIFAR-10 datasets to remove or introduce potential sources of bias does affect the fairness impact of pruning. The effect does not appear to be limited to a single pruning method, but different pruning methods do not experience the effect equally.

The fairness impact of data-driven pruning can be improved through a simple tweak to the cross-entropy loss. The performance weighted loss function assigns weights to samples based on the performance of the unpruned model and uses the corrected output of the unpruned model as classification targets. These small changes improve the fairness of existing pruning methods with some models. The performance weighted loss function does not appear to be universally beneficial, but it is a useful tool for machine learning practitioners who seek to compress models in fairness sensitive contexts.

Acknowledgements

I would first like to thank my supervisor Prof. Alexander Wong for his support and guidance throughout my degree. Your passion is inspiring and you have helped me to grow as a researcher tremendously.

To everyone at the Vision and Image Processing Lab, thank you for your guidance and camaraderie. Having the opportunity to work with, learn from and play board games with you has been one of the highlights of my degree.

I would also like to thank Prof. Chen and Prof. Zelek for taking the time to review my thesis. I greatly appreciate you taking the time to read my thesis.

Finally, I would like to thank the Natural Sciences and Engineering Research Council of Canada, the Ontario Graduate Scholarship and the University of Waterloo for providing me with the funding required to complete my degree.

Dedication

This thesis is dedicated to my parents, who have always been there for me. I would not be here without you.

Table of Contents

Author's Declaration ii

Statement of Contributions iii

Abstract iv

Acknowledgements v

Dedication vi

List of Figures x

List of Tables xii

1 Introduction 1

1.1 Problem Definition . . . 2

1.1.1 Defining Fairness . . . 2

1.1.2 Defining Model Compression . . . 3

1.1.3 Defining Fair Model Compression . . . 4

1.2 Contributions and Outline . . . 4

2 Background 6

2.1 Convolutional Neural Networks . . . 6

2.2 Designing Fair Machine Learning Systems . . . 8

2.3 Convolutional Neural Network Compression . . . 10

2.3.1 Pruning . . . 10

2.4 Fair CNN Compression . . . 12

3 An Empirical Analysis of Fair Pruning 13

3.1 Analysis Protocol . . . 13

3.1.1 Metrics . . . 14

3.2 Pruning Biased CNNs . . . 15

3.2.1 CelebA . . . 15

3.2.2 Fitzpatrick17k . . . 16

3.3 Adjusting the Attribute and Class Balance . . . 18

3.4 Inducing Bias in an Unbiased Dataset . . . 20

3.4.1 Bias source: Underrepresentation . . . 21

3.4.2 Bias source: Visual artifact . . . 22

3.5 Summary . . . 24

4 The Performance Weighted Loss Function 25

4.1 Motivation . . . 26

4.2 Method . . . 26

4.3 Experiments . . . 28

4.3.1 Experimental Set-up . . . 28

4.3.2 Evaluating Fairness and Performance . . . 29

4.3.3 Loss Analysis . . . 33

4.3.4 Ablation . . . 34

4.3.5 Summary . . . 36

5 Conclusion **37**

5.1 Summary of Contributions 37

5.2 Recommendations 38

5.3 Limitations 38

5.4 Future Work 39

5.4.1 Identifying the Properties of Pruning Methods That Affect Fairness 39

5.4.2 Exploring More Sophisticated Methods for Improving Model Fairness 40

5.4.3 Developing a Fair Pruning Benchmark 40

5.5 Discussion 40

References **42**

APPENDICES **47**

A Model Training and Pruning Parameters **48**

List of Figures

2.1 A simple feedforward neural network 6

2.2 A simple convolutional neural network (CNN) 8

2.3 Unstructured and structured pruning of a simple CNN 11

3.1 Sample CelebA images 16

3.2 Sample Fitzpatrick17k images 16

3.3 Mean pruning performance with the CelebA dataset 17

3.4 Mean pruning performance with the Fitzpatrick17k dataset 18

3.5 Mean pruning performance with the CelebA dataset with alternative attribute and class balance 19

3.7 Mean pruning performance with the CIFAR-10 dataset 21

3.6 Sample CIFAR-10 images with and without the visual artifact 21

3.8 Mean pruning performance with the CIFAR-10 dataset with underrepresentation 22

3.9 Mean pruning performance with the CIFAR-10 dataset with visual artifact 24

4.1 Overview of the performance weighted loss (PW Loss) 25

4.2 Mean pruning performance with and without the performance weighted loss function on the CelebA task 30

4.3 Mean pruning performance with and without the performance weighted loss function on the Fitzpatrick17k task 31

4.4 Mean pruning performance with and without the performance weighted loss function on the CIFAR-10 task 32

4.5 Mean pruning performance with and without the performance weighted loss function on the CIFAR-10 with underrepresentation and CIFAR-10 with visual artifact tasks 33

4.6 Change in the proportion of the batch loss due to the use of the PW loss . 34

4.7 Pruning performance with the CelebA dataset when elements of the PW loss are applied independently 35

List of Tables

1.1 Three Definitions of Fairness for Binary Classification 3

3.1 CelebA Training Data Composition 15

A.1 Parameters used for AutoBot pruning method 50
A.2 Parameters used for Taylor pruning method 50
A.3 Parameters used for PW loss 50
A.4 Parameters used to retrain models 51

Chapter 1

Introduction

Deep learning systems have been applied to a variety of computer vision problems with great success. By enabling a level of performance that was otherwise impossible to achieve, deep learning has become a leading approach for computer vision tasks such as image classification [45], object detection [52] and image segmentation [32]. Today, deep learning systems are applied to a wide variety of applications such as medical imaging [41], mobile photography [6] and factory automation [48].

However, the effectiveness of a system depends on more than task performance. Efficiency and fairness are also important considerations. A system with an excessive computational cost may not be able to deployed in a cost effective manner and a system that has poor performance for a subset of users may deepen social inequalities.

While the goals of computational efficiency and fairness may appear to be unrelated, recent research has demonstrated that common methods for compressing deep learning models to improve efficiency can exacerbate existing model biases [16, 36]. There is no guarantee that the impact of model compression will be equal for all potential inputs. Some inputs are more likely to be associated with decreased performance after model compression. In particular groups which are complex or underrepresented are more likely to see deteriorated performance from model compression [36]. The use of model compression could consequently cause deep learning systems to discriminate against particular subgroups.

Discriminatory deep learning systems are not a hypothetical. Organizations such as Amazon and Twitter have reworked resume scanning [5] and automatic cropping systems [31] due to fairness concerns. Governments and regulators are also starting to take notice of the issue. The White House specifically addresses the problem of fairness in AI systems in their "Blueprint for an AI Bill of Rights" [17].

To avoid such issues, machine learning practitioners must consider the fairness impact of each design decision they make in the creation of their system. Model compression decisions are no exception. To use model compression practitioners should understand and know how to manage the fairness impacts of model compression. In fairness sensitive contexts such as healthcare, personnel management and security, using model compression without considering the fairness impacts of compression could have significant social effects.

In this work, we aim to examine the relationship between model compression and fairness for deep learning vision systems. In particular, we focus on convolutional neural networks, the dominant deep learning approach for vision problems, and model pruning, a highly popular approach for model compression.

1.1 Problem Definition

1.1.1 Defining Fairness

As fairness is a social concept there is no single formal mathematical definition for fairness. At a high-level, fairness can be conceptualized as the idea that model performance should be independent of a selected attribute. The attribute in question is context specific and could correspond to social identifiers such as class, race and gender.

The particular fairness definition used for an application depends on the social context of the application. In some situations it may be desired that the system has no significant discrepancies in behaviour between all attribute subgroups. In other situations, parity in a few key metrics may be sufficient.

To illustrate this, we can consider a few different definitions of fairness in binary classification. As defined in Table 1.1, demographic parity, equalized odds and equal opportunity are three simple notions of fairness that can be applied to binary classification [11]. Demographic parity simply states that a decision, $\hat{Y}$, is fair if it is independent of the attribute A [11]. However, demographic parity is flawed in many circumstances as the ideal probability of positive classification can vary between attribute subgroups. Equalized odds allows for correlations between the attribute and the classification target by instead stating that when given a classification target, Y, the correctness of a fair decision is independent of the attribute [11]. Equal opportunity softens the equalized odds requirement by stating that when given a positive classification target, the correctness of a fair decision is independent of the attribute [11]. If equalized odds holds true, the accuracies of the sample groups with and without the attribute will be equal. If equal opportunity holds true, the true positive rate will be equal for sample groups with and without the attribute.

Table 1.1: Three Definitions of Fairness for Binary Classification

Name	Definition	
Demographic Parity	$Pr\{\hat{Y} = 1 \mid A = 0\} = Pr\{\hat{Y} = 1 \mid A = 1\}$	
Equalized Odds	$Pr\{\hat{Y} = 1 \mid A = 0, Y = y\} = Pr\{\hat{Y} = 1 \mid A = 1, Y = y\}$	$y \in \{0, 1\}$
Equal Opportunity	$Pr\{\hat{Y} = 1 \mid A = 0, Y = 1\} = Pr\{\hat{Y} = 1 \mid A = 1, Y = 1\}$	

In contexts in which the effects of the system are primarily attributed to positive classification, such as university admissions or security monitoring, equal opportunity may be preferred over equalized odds as it more closely captures the impact of the system. Ultimately, it is the impact of the machine learning system that is of concern. Performance imbalances that have no impact on users are not fairness concerns. An appropriate fairness definition mus therefore consider the impact of a system's output within the social context of the system.

Understanding fairness impacts without a particular system in mind can be difficult. For this thesis, our approach is to study the overall fairness impact of compression with the understanding that all of our findings may not be applicable to every system.

1.1.2 Defining Model Compression

Model compression is a process by which the size of a neural network can be reduced to reduce computational cost. There are generally two goals of compression: reducing the memory footprint of the network and reducing the inference time of the network. Performant model compression approaches are able to achieve these objectives with minimal impact to model performance.

There are many different compression approaches that have been proposed. Some approaches focus on the number of model parameters while other approaches focus on the size of each parameter [34]. Practitioners do not have to use a single approach to compress a network. Instead, many compression approaches can be used in concert.

For this work, we focus on model pruning. Model pruning is a popular compression approach in which parameters are removed from a model to reduce the size of the model

[34]. Not all parameters in a neural network are equally important. Some parameters may be redundant or unused, and have little impact on the model's behaviour. Pruning approaches seek to identify and remove these parameters.

1.1.3 Defining Fair Model Compression

For this work, we define fair model compression as the performant compression of a neural network in a fair manner. In contrast to the model compression problem which emphasizes the preservation of the overall performance of the mode, the fair model compression problem also considers the preservation of fairness. A fair model compression approach is able to reliably compress a model to a specified degree of compression with minimal impact to both overall performance and fairness.

As fairness has a context specific definition, so too does fair model compression. The specific definition of fairness as well as the desired balance between overall performance and fairness depend on context and goals of the system.

In this thesis, we consider a compression method unfair if the decrease in performance experienced by samples in a subgroup is greater than the decrease experienced by samples not in the subgroup. We measure performance using the area under the receiver operating curve (ROC-AUC). As it is a threshold agnostic performance metric, the ROC-AUC is a good measure of the model's understanding and separability for a subgroup [2]. Fair compression methods will induce similar changes in the ROC-AUC for all subgroups.

1.2 Contributions and Outline

This thesis aims to explore the problem of fair model compression by examining the existing compression methods and developing a new compression method. The primary contributions of this thesis are an analysis of the fairness of existing compression methods and the development of a method for improving the fairness of pruning approaches for model compression. The goal of these contributions is to develop an understanding of the mechanism by which unfairness can arise during model compression and to apply this understanding to improve fairness during model compression.

Chapter 2 of this thesis provides background information on model compression and fairness in machine learning. Chapter 3 presents a quantitative analysis of the fairness of a selection of model pruning methods. Chapter 4 introduces a novel loss function that can improve the fairness of existing pruning methods. Chapter 5 concludes the thesis by reflecting on the findings of the previous chapters, making recommendations and identifying future work.

Chapter 2

Background

2.1 Convolutional Neural Networks

Convolutional neural networks (CNNs) are a form of neural networks that are designed to process images. By introducing an operation that is similar to the convolution into the structure of a neural network, CNNs are able to effectively and efficiently process images. Today, CNNs are used for a wide variety of tasks including image classification, object segmentation and more [25].

As depicted in Figure 2.1, a basic feedforward neural network contains layers of interconnected nodes. Each node in the first layer uses the system inputs as its input values while the nodes of every subsequent layer uses every node in the previous layer as its input values. The output value of every node is calculated by applying a simple non-linear transformation to a linear combination of its input values. The coefficients of the linear transformation, known as weights, determine the behaviour of the network while the non-linear transformation enables the network to model non-linear behaviour.

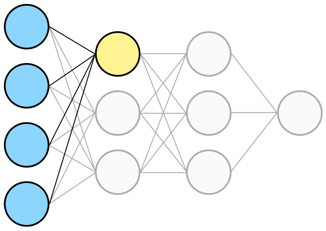


Figure 2.1: A simple feedforward neural network. The value of each node in the network depends on the values of all nodes in the previous layer.

CNNs extend the feedforward neural network by replacing the fully connected layers of the feedforward network with convolutional layers. A basic CNN structure is depicted in Figure 2.2. Convolutional layers multiply and sum rectangular regions of an input image with a filter in a sliding window fashion. Rather than connecting each node to all nodes in the previous layer, nodes in convolutional neural networks are only connected to the nodes found in a single rectangular region of the previous layer. Furthermore, as the filter is applied to the whole image using a sliding window approach, filter weights are shared across the entire input. These differences allows a convolutional layer to process an input of a given size more efficiently than a fully connected layer.

The differences also impose two inductive biases. The use of continuous rectangular regions assumes that related values can be found near each other. The use of constant filters applied with a sliding window approach assumes that useful patterns can manifest identically in all regions of the input. These assumptions are typically relevant for natural images as objects depicted in images are typically located in continuous regions that can be anywhere in the image. CNNs are consequently well suited for image processing tasks.

A CNN is trained to complete a specific task by finding the parameter values that minimize a specified loss function for a training dataset. An iterative optimization approach such as stochastic gradient descent is used to find the optimal parameter values. For classification, the average cross-entropy is typically minimized. The cross-entropy, l_{CE}, is shown in equation 2.1, where M is the number of classes, y_i is the true probability for the ith class and $\hat{y}_i$ is the predicted probability for the ith class.

$$l_{CE} = -\sum_{i=1}^{M} y_i \log(\hat{y}_i) \tag{2.1}$$

A typical CNN contains many convolutional layers followed by a small number of fully connected layers. CNN architectures vary in properties such as the number of layers, the number of filters in each layer and the size of each filter. Modern architectures also introduce additional features such as residual layers that sum the output values of multiple layers [13] and depth-wise separable convolutions which learn separate filters for each input image channel [18]. However, all CNN architectures use the convolution as the primary processing step. Convolutional layers contribute almost all of a CNN's parameters and almost all inference operations are related to convolutional layers.

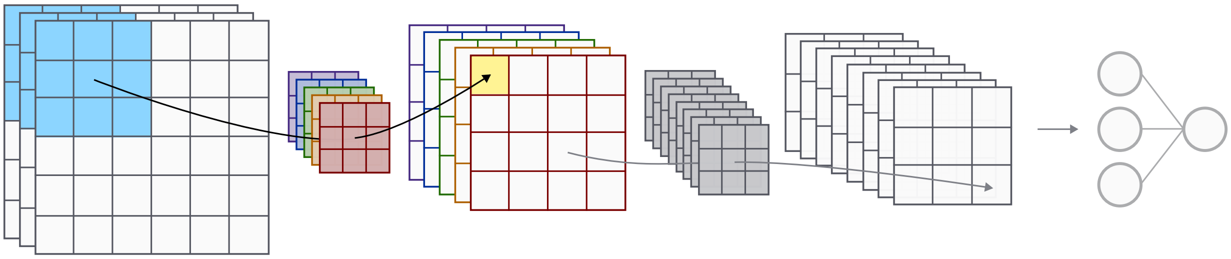


Figure 2.2: A simple convolutional neural network (CNN). The value of each node depends on all nodes found in a region within the previous layer. Furthermore, each output channel is formed using a single convolutional filter that is applied to the previous layer in a sliding window fashion.

2.2 Designing Fair Machine Learning Systems

Machine learning (ML) systems are complex software systems that fit a model to a training dataset. Through the fit procedure, the system learns how the desired behaviour can be achieved. As the behaviour of the model is learned rather than specified, it can be difficult to predict exactly how the system will perform. The model may adopt an approach that works well for most inputs, but performs poorly for inputs from specific subgroups. In other words, the model may learn unfair behaviour.

Fairness issues have been observed in a number of real world systems. Chest X-ray screening systems [40], facial recognition systems [43] and resume screening systems [5] have all been show to have biases. We can not simply assume that the ML systems we develop will be fair. Instead we must carefully develop our ML systems with fairness in mind. To this end, the field of fair machine learning aims to understand and mitigate fairness issues in ML systems.

As described in Section 1.1.1, the precise definition of fairness depends on the context of the system. Likewise, the precise methods required to design fair machine learning systems also depend on the context of the system. Introducing auditing mechanisms to identify fairness issues may be sufficient for one system while the prevention of fairness issues may be a requirement for a different system.

As such, there are a diverse range of fair ML approaches that have been proposed. Different approaches target different areas within the machine learning system using different methods. To design a fair machine learning system that is simultaneously performant and robust to fairness issues, multiple approaches may need to be combined.

We can group the available approaches by the system area they directly address. Fair ML approaches have been developed that address the system's data pipeline, model design, fitting procedure and feedback protocol.

Approaches that modify the system's data pipeline seek to mitigate fairness issues at the source. These approaches may generate new training samples or transform existing training samples to improve the fairness of the data pipeline. For instance, BiaSwap [23] automatically generates new training images by merging bias-relevant details from one image with bias-irrelevant details from another image. FARE [22] maps the inputs to an encoding space with a provable upper-bound on the unfairness of a downstream classifier.

Model design approaches seek to improve fairness by adjusting the structure of the model. Unfairness in ML systems is typically learned rather than designed. However, the model design still controls how the model can be used, influencing system fairness. For example, Norrenbrock, Rudolph, and Rosenhahn [35] propose the use of a model with a sparse and low-dimensional final decision layer to improve interpretability and enable human auditing of the model's decision making process.

The fitting procedure determines how data is applied to select a single solution within the solution space defined by the model design. Approaches that address the fitting procedure seek to encourage the procedure to select a fair solution. These approaches may make modifications such as introducing additional fitting stages or augmenting the loss function with fairness criteria. Zhang et al. [49] propose an additional pre-processing stage that uses unlabelled data to improve fairness. Jain, Huber, and Elmasri [20] characterize fairness using a Bias Parity Score and incorporate this score into the system's loss function.

Approaches that modify the feedback protocol seek to improve system fairness by allowing fairness issues to be identified quickly. Swift and reliable fairness auditing may not prevent fairness issues from arising, but it can identify unfair machine learning systems that should not see continued use without modification. Many auditing frameworks [3, 38, 39] have been proposed to systematically identify any potential fairness issues in machine learning systems.

While each of these approaches aim to improve the fairness of machine learning systems, the way in which they do so varies greatly. Different combinations of approaches may be optimal for different systems. To build fair systems, machine learning practitioners need to analyze their system to determine which approaches would be most beneficial. This can be challenging as machine learning systems are complex with many potential sources of bias. The specific fairness impact of each machine learning design decision is not well understood. Research into the sources of bias in machine learning systems and the approaches that best address each source could help practitioners to design fair machine learning systems.

2.3 Convolutional Neural Network Compression

Convolutional neural networks are powerful tools that have demonstrated real-world usefulness in many applications. However, CNNs contain millions of parameters and execute millions of operations for each inference. Many large CNNs are simply unable to be used in computationally constrained environments such as mobile phones and smart home devices.

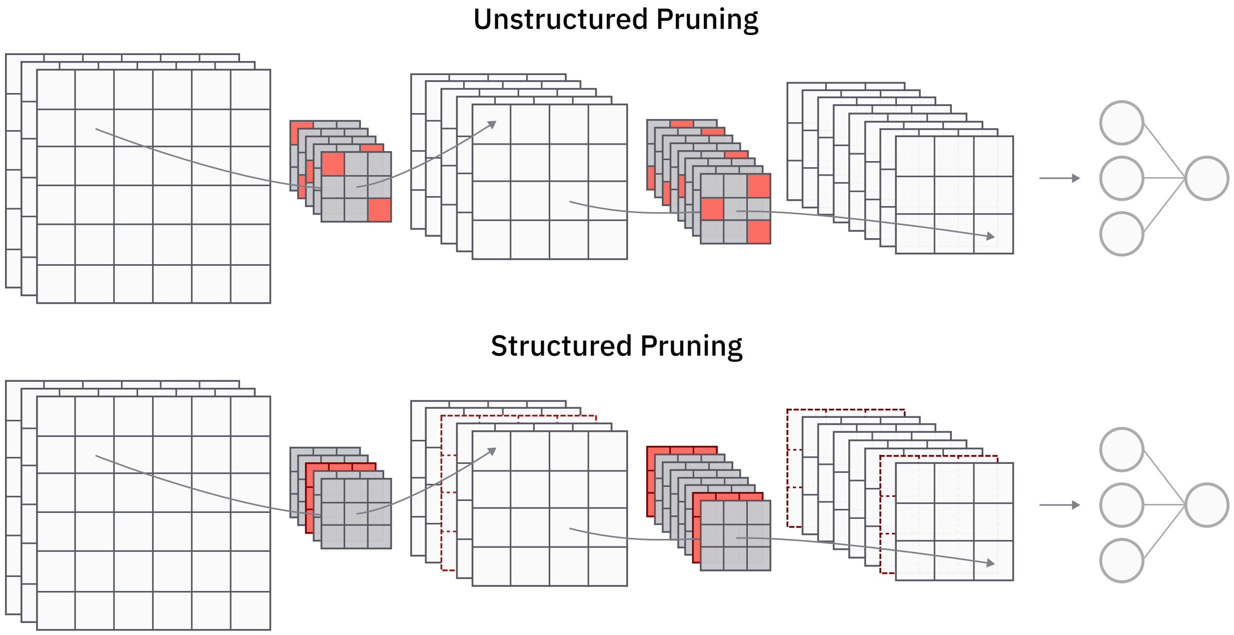
CNN compression is the practice of transforming a large CNN into a small CNN. High quality model compressions produce small models that performs similarly to the large original model. It has been demonstrated that small CNNs that are the result of compression generally outperform CNNs of a similar size that are trained directly [50]. Not all networks of equal size are equivalent. By training a large network to find a good solution and then compressing the solution, we can build a small model that is tailored to the selected task.

There are a variety of compression approaches that have been proposed. Pruning, quantization, weight-sharing and tensor decomposition approaches are some of the more common compression approaches [34]. Pruning approaches remove redundant and unused parameters from the network. Quantization approaches reduce the precision of the parameters to reduce the size of each parameter. Weight sharing approaches reuse parameters to reduce the number of independent parameters. Tensor decomposition approaches transform parameter tensors into near-equivalent but smaller low-rank representations. In this work, we focus on network pruning as it is a highly popular method that is applicable to most CNNs.

2.3.1 Pruning

Neural network pruning removes parameters from the network to reduce the size of the network. By reducing the number of parameters, pruning approaches directly reduce the memory requirements of the network and the number of operations required for each interference. Alternatively, pruning can be thought of as the identification of a subnetwork that approximates the full network.

As illustrated in Figure 2.3, pruning approaches can be divided into two categories: structured and unstructured [15]. Unstructured approaches prune individual parameters whereas structured approaches prune groups of parameters. The solution space of structured approaches is constrained, but structured pruning approaches are often more practical than unstructured approaches. By pruning groups of parameters, entire high-level operations such as matrix multiplications and convolutions can be skipped, improving inference times with standard hardware and software libraries [15]. Structured pruning of CNNs typically involves pruning entire convolutional filters.



Unstructured Pruning

Structured Pruning

Figure 2.3: Unstructured and structured pruning of a simple CNN. Entire convolutional filters are pruned in the structured example. This leads to easily realizable computational gains as entire output channels no longer need to be computed.

Many pruning approaches have been proposed, each with a unique method of selecting parameters to pruned. While simple approaches may only consider the parameter values, other approaches use data to identify the least influential parameters. These approaches often use gradient information or score each parameter through an optimization process.

Molchanov et al. [33] propose the use of a scoring metric based on a Taylor expansion that uses gradient information to identify filters to prune. Discrimination-aware channel pruning [51] uses gradient information from a loss function that contains a cross-entropy term and a reconstruction error term. Ding et al. [7] propose ResRep, a method that involves re-parameterizing the network into ‘remembering’ and ‘forgetting’ parts and training each part using stochastic gradient descent with appropriate criteria. Castells and Yeom [4] propose AutoBot, a method that inserts trainable bottlenecks into the network to selectively restrict the flow of information to the parameters.

As pruning can significantly alter a network, most pruning approaches retrain the network after pruning. Some approaches repeatedly alternate pruning with retraining while others use a single pruning step followed by a single retraining step.

2.4 Fair CNN Compression

While there is significant research in the fields of fair ML and CNN compression, existing research into fair CNN compression is limited. Nevertheless, there is some existing research into the relationship between compression and fairness in neural networks.

Researchers have established that compression can impact fairness and identified some approaches to managing the fair compression problem. Hooker et al. [16] propose auditing samples affected by model compression, called Compression Identified Exemplars, as an approach for identifying and managing the negative effects of model compression. Paganini [36] demonstrates how class imbalances and differences in class complexity can cause pruning approaches that only consider overall accuracy to reduce class fairness. Joseph et al. [21] propose a multi-part loss function intended to improve the alignment between predictions between the original and pruned model. They demonstrate that their method can have beneficial effects with respect to class-wise fairness.

Additionally, other papers have examined how compression methods can be used to improve model fairness. While these papers are not addressing the problem of fair CNN compression as defined in this work, they demonstrate the relationship between compression and fairness. Wu et al. [46] propose Fairprune, a method for improving model bias using pruning. Fairprune prunes parameters using a saliency metric to increase model fairness. Xu and Hu [47] propose the use of knowledge distillation and pruning to reduce bias in natural language models. Marcinkevičs, Ozkan, and Vogt [30] propose a debiasing procedure that involves pruning parameters using a gradient based influence measure.

All of these findings firmly establish that compression and fairness are related. However, the nature of this relationship is still largely unknown due to the sparsity of available research. It is not known how the problem could manifest in various environments and very few approaches for managing the fairness impact of compression have been explored. More research is needed to equip practitioners with the tools they need to consistently compress CNNs in a fair manner.

Chapter 3

An Empirical Analysis of Fair Pruning

Only a small number of papers have examined the effect of pruning on fairness. Paganini [36] performed a series of experiments that demonstrated that pruning could have a detrimental effect on subgroups that are complex or underrepresented in training data. However, their analysis focused solely on class-wise fairness in datasets without known fairness issues. Hooker et al. [16] demonstrated that pruning exacerbated biases for a model trained with the CelebA dataset. However, their analysis was limited to a single model trained on a single dataset and pruned with a single pruning method.

In this chapter, we examine the effect of filter pruning on a variety of convolutional neural networks trained using a variety of datasets. We examine the effect of pruning on networks with known biases using three different pruning protocols. We then extend this analysis by manipulating the training data to eliminate possible sources of bias and introduce potential sources of bias into a dataset without known biases.

3.1 Analysis Protocol

We evaluated three different filter pruning methods. The first method is AutoBot [4], an accuracy preserving pruning method that uses trainable bottleneck parameters that limits the flow of information through the model. The second method uses an importance metric derived from the Taylor expansion of the loss function [33]. The third method is a simple random pruning protocol that randomly selects filters to prune.

As it is simply infeasible to evaluate all of the numerous pruning approaches that have been proposed in literature, we selected three different pruning methods that represent the pruning landscape. AutoBot is a novel pruning method that uses an optimization process to identify the parameters to prune. It makes few assumptions about the structure of the pruning network or the properties of parameters that should be pruned. The Taylor method of Molchanov et al. [33] is an widely cited pruning method that iteratively prunes filters using a derived metric that incorporates gradient information. This is an approach that many new methods have extended. The random pruning method is not an approach that we would expect to be used in real-world situations. However, it is useful as a baseline against which the other methods can be compared.

We implemented the methods using the *PyTorch* library [37]. The methods were implemented as three step pipelines in which the model is first pseudo-pruned by setting parameters to zero, fully pruned using the *Torch-Pruning* library [9] and retrained. Pseudo-pruning allows for fast pruning during the pruning process while the full pruning step removes the unused parameters, reducing the number of operations required for prediction. Due to dependencies between parameters introduced by structures such as residual layers, the achieved theoretical speedup can differ from the target theoretical speedup. All hyperparameters for the pruning methods were selected using a hold-out validation set. We repeated each experiment three times. All figures displaying model performance after pruning are displaying the average of all trials.

3.1.1 Metrics

The methods were evaluated by pruning models with various target theoretical speedups, defined as the number of floating point operations (FLOPS) of the original model divided by the FLOPS of the pruned model. As our primary concern is the degradation of a model's overall behaviour towards different subgroups due to pruning, we compared the change in the areas under the receiver operator curves (ROC-AUC) for relevant subgroups.

The ROC-AUC is a threshold agnostic metric that is calculated by measuring the true positive and false positive rates of the model at various classification thresholds and measuring the area under the resultant curve. A perfect classifier would have an ROC-AUC of 1, while a random classifier would have a ROC-AUC of 0.5. By measuring the ROC-AUC for each subgroup we can measure the model's understanding of each subgroup [2]. We calculate the change in subgroup ROC-AUC by measuring the ROC-AUC of the model before and after pruning for a subgroup and subtracting the unpruned ROC-AUC from the pruned ROC-AUC. For non-binary classification we used the average one-vs-rest ROC-AUC.

3.2 Pruning Biased CNNs

We tested all of our selected pruning methods using two different classification tasks involving datasets with known biases.

Our first task was the celebrity face classification task using the CelebA dataset [27] as outlined by Hooker et al. [16], in which a model is trained to identify faces as blonde or non-blonde. The CelebA dataset contains over 200 000 images of celebrity faces with various annotations. Sample CelebA images can be found in Figure 3.1 and the composition of the CelebA training data is described in Table 3.1. While blonde non-male samples make up 14.05% of the training data, blond male samples make up only 0.85% of the training data. We used the provided data splits with 80% of the available data being used for training with the remaining data split evenly for validation and testing.

Our second task was the skin lesion classification task using the Fitzpatrick17k dataset [10]. The Fitzpatrick17k dataset consists of 16 577 images of skin conditions. Sample Fitzpatrick17k images can be found in Figure 3.2. We trained our models to classify the samples as non-neoplastic, benign or malignant. Due to missing and invalid images we were only able to use 16 526 images. Each sample in the dataset is assigned a Fitzpatrick score that categorizes the skin tone of the sample. We trained our models on only samples with light skin tone scores of 1 or 2, and evaluated the model on medium skin tone scores of 3 or 4 as well as dark skin tone scores of 5 or 6. We used a random 25% of the medium and dark skin tones as a validation set with the remainder used as a test set. As the skin tones the model is trained on and evaluated on differ, this is an out-of-distribution task.

Table 3.1: CelebA Training Data Composition

	Male	Non-Male
Blonde	1387	22880
Non-Blonde	66874	71629

3.2.1 CelebA

We trained a ResNet-18 [14] model and a VGG-16 [42] model for the CelebA task. The ROC-AUCs for the male and non-male subgroups of the ResNet-18 model were 0.9639 and 0.9794 respectively. The ROC-AUCs for the male and non-male subgroups of the VGG-16 model were 0.9679 and 0.9825 respectively. Both models were pruned using target theoretical speedups of 16, 32, 64, 128 and 256.

Figure 3.1: Sample CelebA images

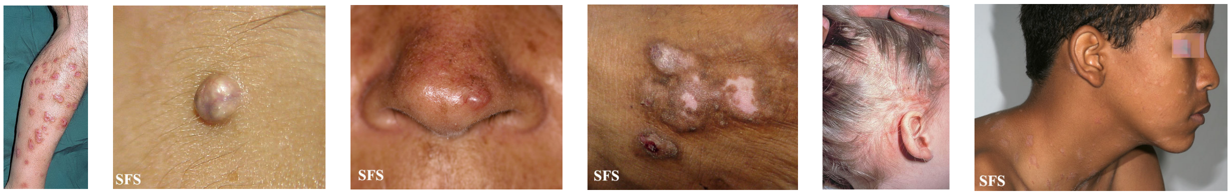


Figure 3.2: Sample Fitzpatrick17k images

The change in ROC-AUC for the pruning methods for the ResNet-18 and VGG-16 models can be found in Figure 3.3. Some of the VGG-16 models pruned using the AutoBot random methods always predicted a single class. These degenerate models were not included in the mean values plotted in the graph.

All methods, including the random method, were able to significantly reduce the size of both models. However, they also caused divergent performance between the male and non-male subgroups. For all tested methods, male samples experienced a larger degradation in performance than non-male samples. This difference was significantly larger for the ResNet-18 model than the VGG-16 model. The difference appears similar for all methods with the ResNet-18 model but the Taylor method appears to be more fair and higher performing with the VGG-16 model. Furthermore, for all methods, the difference is larger at higher theoretical speedups.

These results indicate that pruning did exacerbate fairness issues. The magnitude of the issue appears to be dependent on both the model and the pruning method used. Nevertheless, we observed this phenomenon with all pruning methods, even random pruning. As the random pruning method was affected, the process of selecting the parameters to prune can not be the sole source of unfairness in the pruning process.

3.2.2 Fitzpatrick17k

We trained a ResNet-34 [14] model and a EfficientNet-V2 Medium [44] model for the Fitzpatrick17k task. The ROC-AUCs for the medium and dark subgroups of the ResNet-

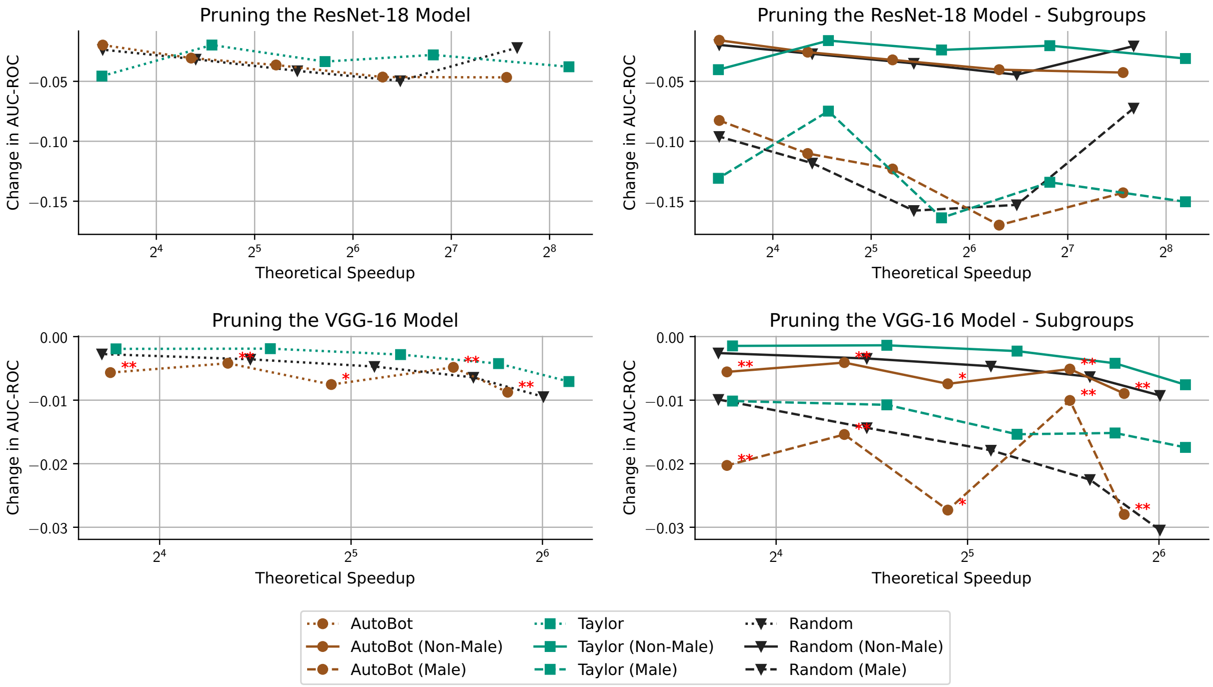


Figure 3.3: Mean pruning performance of the ResNet-18 and VGG-16 models with the CelebA dataset. A red star next to a data point indicates that a degenerate model was excluded from that data point.

34 model were 0.8190 and 0.7329 respectively. The ROC-AUCs for the medium and dark subgroups of the EfficientNet model were 0.8516 and 0.7524 respectively. Both models were pruned using target theoretical speedups of 2, 4, 8, 16 and 32.

The change in ROC-AUC for all tested pruning methods for the Fitzpatrick17k models can be found in Figure 4.3. Despite a bias against dark skin tones existing in the original models, we do not see divergent AUC-ROC scores as the theoretical speedup increases. The medium skin tone subgroup actually saw greater changes in AUC-ROC due to pruning.

Pruning does not appear to be guaranteed to exacerbate existing fairness issues. While the unpruned models were biased against dark skin tones, pruning decreased performance of the medium skin tone subgroup more than the dark skin tone subgroup. Therefore, in certain situations pruning may actually improve fairness.

The difference between these results and the pruning results of CelebA task may be due to the difference in bias between the two datasets. The CelebA models performed poorly with the male subgroup as the label distribution of the male subgroup was highly skewed in the training data. In contrast, neither medium nor dark skin tones were included in the training data for the Fitzpatrick17k models. Instead, the bias against dark skin tones is likely due to the medium skin tones resembling more closely resembling the light skin tones used for training. It is possible that performance penalties arising from a difference in training and testing sample distributions may not be aggravated by pruning.

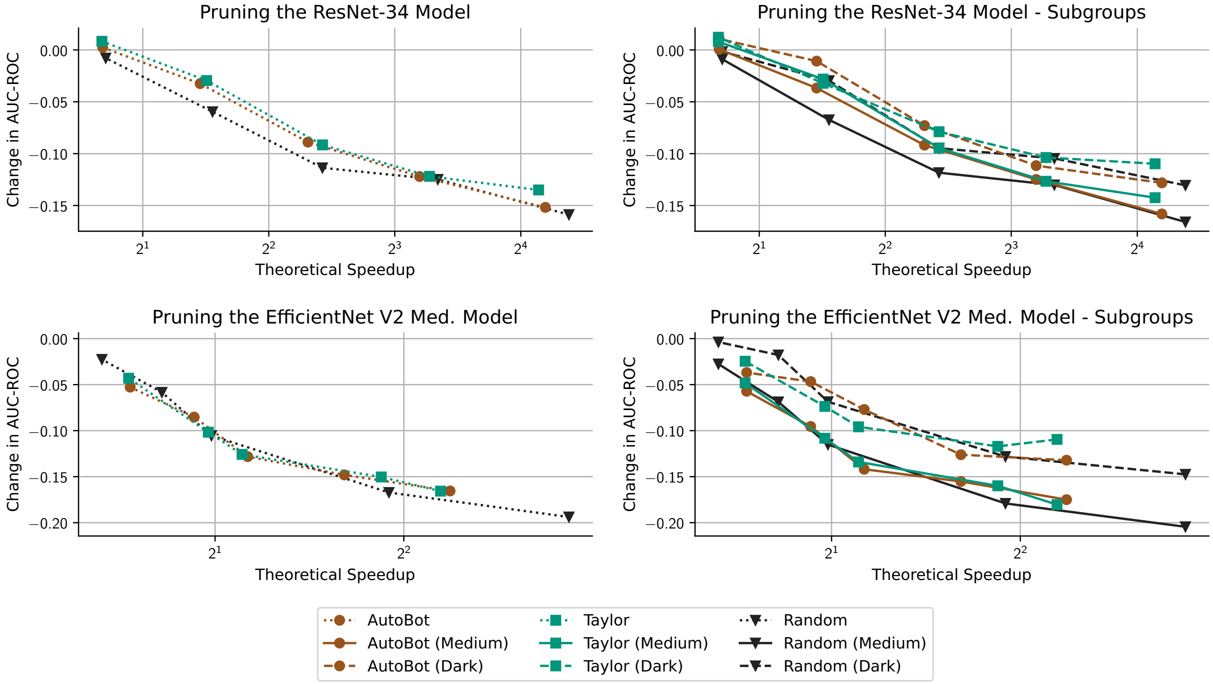


Figure 3.4: Mean pruning performance with ResNet-34 and EfficientNet V2 Med. models with the Fitzpatrick17k dataset.

3.3 Adjusting the Attribute and Class Balance

Our CelebA results demonstrated that pruning can exacerbate existing biases. However, it is not clear which properties of the dataset cause this phenomenon. One simple possible cause is the underrepresentation of blond male samples in the dataset. To test this hypothesis, we trained models using balanced subsets of the dataset.

We created three artificial datasets from the CelebA dataset by selected subsets of the training data. The first subset was formed using 3.41% of the available training data such

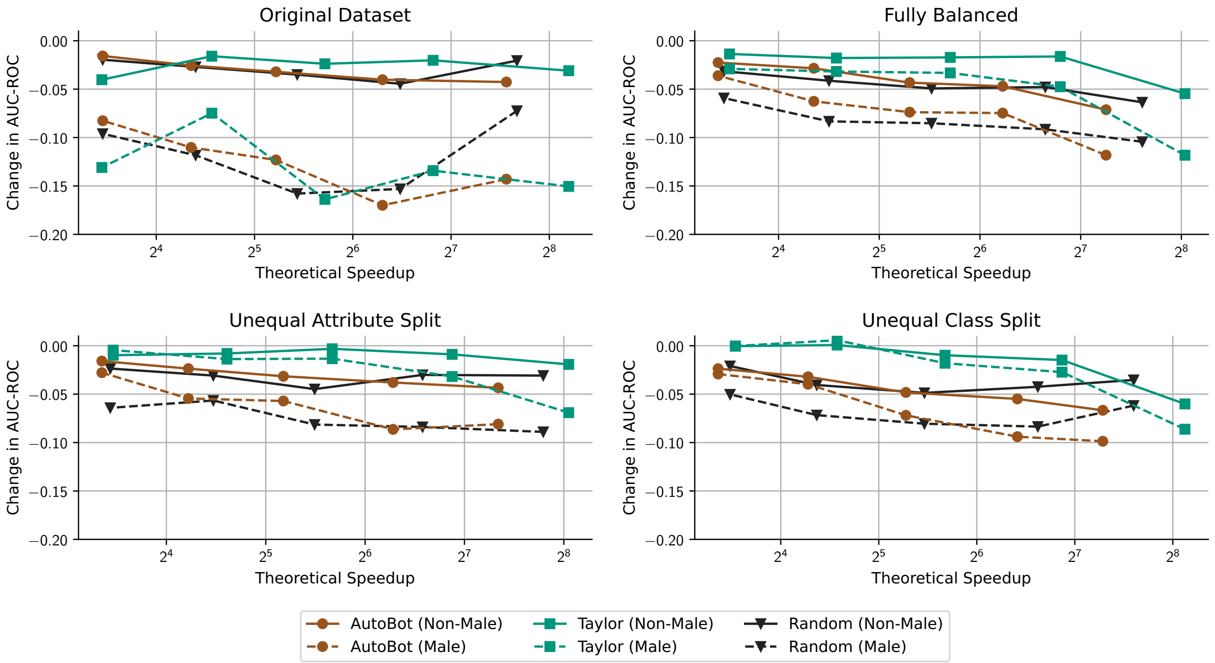


Figure 3.5: Mean pruning performance of the ResNet-18 model with the CelebA dataset with alternative attribute and class balances.

that it was fully balanced, containing an equal number of male and non-male samples as well as an equal number of blonde and non-blonde samples. The second and third subsets were formed by adding additional samples to the first subset, altering the attribute or class balance. The second subset contained an equal number of blonde and non-blonde samples, but five times as many non-male samples as there were male samples. The third subset contained an equal number of male and non-male samples, but five times as many non-blonde samples as there were blonde samples. The full unmodified test set was used for all subsets.

A ResNet-18 model was trained using each subset. Compared to the original model, these new models had moderately lower performance for both the male and non-male subgroups. The AUC-ROCs for the male subgroup are 0.9562, 0.9479 and 0.9183 for the first, second and third subsets respectively. The AUC-ROCs for the non-male subgroup are 0.9713, 0.9732 and 0.9580 for the first second and third subsets respectively. The models were pruned using the AutoBot and Taylor methods using target theoretical speedups of 16, 32, 64, 128 and 256. The performance after pruning for these models can be found in Figure 3.5.

The difference in performance after pruning between subgroups is smaller for all three balanced datasets than it is for the original dataset. The Taylor pruning method saw the greatest improvement in fairness due to improving class balance, but all pruning methods did see benefits. For the AutoBot method, the results were similar across all three balanced datasets. By a small margin, the model trained with dataset with an unequal class balance but an equal attribute balance appears to have the fairest pruning results.

All three methods of balancing the data appeared to have a similar impact on fairness. The unequal attribute balance of the second artificial dataset and the unequal class balance of the third artificial dataset did not appear to have a significant impact on fairness. The fairness issue observed when pruning the model trained the full CelebA dataset is likely caused by the interplay of an unequal attribute distribution and an unequal class distribution.

Improving the attribute and class balance did appear to improve fairness after pruning. However, male samples still experienced a greater decrease in AUC-ROC due to pruning than non-male samples. These results indicate that the dataset composition does influence, but not fully explain, the fairness impact of model pruning.

3.4 Inducing Bias in an Unbiased Dataset

To further identify the causes of the fairness impact of pruning, we attempted to induce the effect in a dataset with no known biases. We utilized the CIFAR-10 classification task [24] for this purpose. The CIFAR-10 dataset consists of 50 000 training images and 10 000 testing images evenly distributed between 10 classes. We trained a ResNet-56 [14] model to classify each testing images into the appropriate class. The ROC-AUC of the model is 0.9957.

We first pruned the model using our three pruning methods. These results can be found in Figure 3.7. All three pruning methods were able to prune the model effectively. We then retrained and repruned the model using two different manipulated CIFAR-10 datasets with potential sources of bias. Our first source of bias was a class underrepresentation while our second source of bias was a visual artifact that correlated with the image class. Sample images with and without the artifact can be found in Figure 3.6.

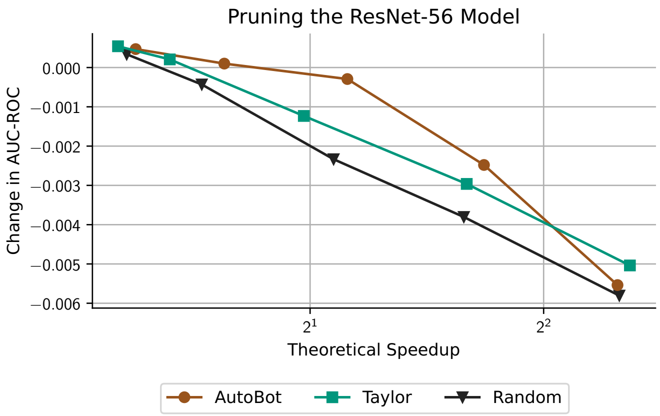


Figure 3.7: Mean pruning performance of the ResNet-56 model with the CIFAR-10 dataset.

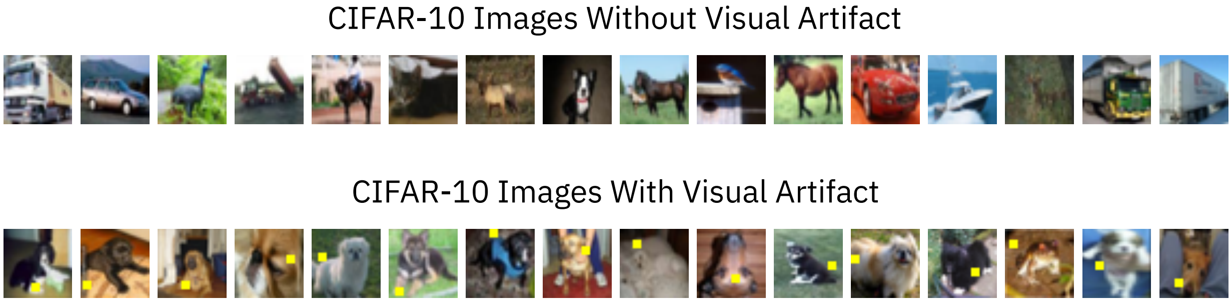

CIFAR-10 Images Without Visual Artifact
CIFAR-10 Images With Visual Artifact

Figure 3.6: Sample CIFAR-10 images with and without the visual artifact

3.4.1 Bias source: Underrepresentation

To test if underrepresentation affects fairness during pruning we removed 80% of the samples associated with the cat and dog classes from the training set. We then trained a new ResNet-56 model. This model had a ROC-AUC of 0.9948. The average one-vs-rest ROC-AUC of the affected classes was 0.9880. The average one-vs-rest ROC-AUC of the affected classes for the original ResNet-56 model was 0.9902.

We then pruned the new model using all three pruning methods and measured both the overall ROC-AUC and the ROC-AUC of the affected classes. The results from these experiments can be found in Figure 3.8.

At lower theoretical speedups the difference between the ROC-AUC of the affected classes and the overall ROC-AUC is small for both the original model and the model trained on the data with underrepresentation. However, as the theoretical speedup increases, the difference grows. This effect can be seen in the results for both of the models.

For the Taylor and random methods this divergence is slightly smaller for the model trained on the data with underrepresentation than it is for the original model. For the AutoBot method at larger theoretical speedups, this divergence is larger for the model trained on the data with underrepresentation. It is difficult to identify precisely why the AutoBot method was negatively affected by the underrepresentation, however we hypothesize that the AutoBot approach is more sensitive to dataset composition as it imposes very few inductive biases and prunes convolutional filters in a one-shot manner.

These results indicate that underrepresentation can worsen fairness issues during pruning, but it is not guaranteed to do so. Introducing underrepresentation into the dataset had a different effect for different pruning methods. The pruning method selected appears to affect the impact of underrepresentation on fairness during pruning.

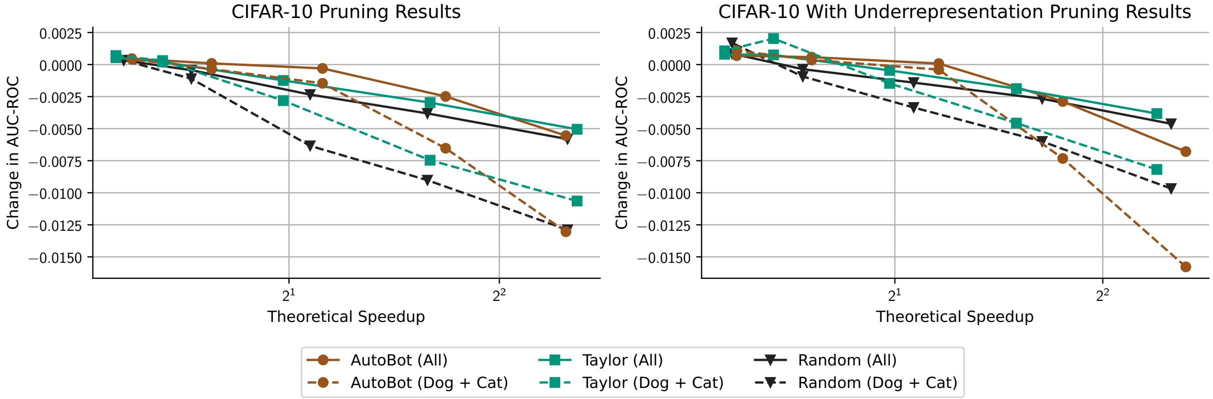


Figure 3.8: Mean pruning performance of the ResNet-56 model with the CIFAR-10 dataset with underrepresentation.

3.4.2 Bias source: Visual artifact

To test how the content of the image can affect fairness during pruning we introduced a visual artifact into some of the training images. Images in the dog class had a 98% chance of having a yellow square placed in the image at a random position. Images in other classes had a 2% chance of containing the yellow square. If a yellow square is found in an image, the image has a 84.5% chance of being associated with the dog class. This value can be calculated using Bayes' Theorum, as shown in equation 3.1 in which S denotes containing a square and D denotes being part of the dog class. While the square may be a simple feature for a classifier to detect, high performing classifiers should not depend on it.

$$P(D|S) = \frac{P(S|D)P(D)}{P(S|D)P(D) + P(S|D^{\complement})P(D^{\complement})} = \frac{0.98 \times 0.1}{0.98 \times 0.1 + 0.02 \times 0.9} = 0.845 \tag{3.1}$$

We trained a new ResNet-56 model on the CIFAR-10 dataset with the visual artifact. This model had a ROC-AUC of 0.9957. The one-vs-rest ROC-AUC of the dog class was 0.9914. We then pruned the new model using all three pruning methods and measured both the overall ROC-AUC and the ROC-AUC of the dog class with and without the visual artifact. The results from these experiments can be found in Figure 3.9.

For all pruning methods, the difference between the overall AUC-ROC and the AUC-ROC of the dog class was small at all tested theoretical speedups when the original model was pruned. In contrast, for all pruning methods with the model trained with the visual artifact, the difference between the overall AUC-ROC and the AUC-ROC of the dog class increased as the theoretical speedup increased. Introducing the visual artifact affected each pruning methods ability to maintain the performance of the dog class. Furthermore, while all pruning methods experienced this effect, they did not experience it equally. The Taylor method was more affected than the AutoBot method.

Interestingly, the AUC-ROC of the dog class with the visual artifact was affected by pruning in a similar manner as the AUC-ROC of the dog class without the visual artifact. This indicates that the pruned models were not simply using the visual artifact as a shortcut to bypass the need to maintain knowledge of more complex features. If this were the case, we would expect performance to not deteriorate as significantly when the artifact is included.

In natural images, the presence of an easy to detect feature that correlated with the classification target could have the same effect as the yellow square. This may explain the fairness issue observed with the CelebA dataset. The male property strongly correlates with the classification target of blonde hair. This correlation could have a similar effect as the correlation of the visual artifact. The pruning results of the CIFAR-10 model trained with the visual artifact does appear similar to the pruning results of the CelebA models as all pruning methods were affected.

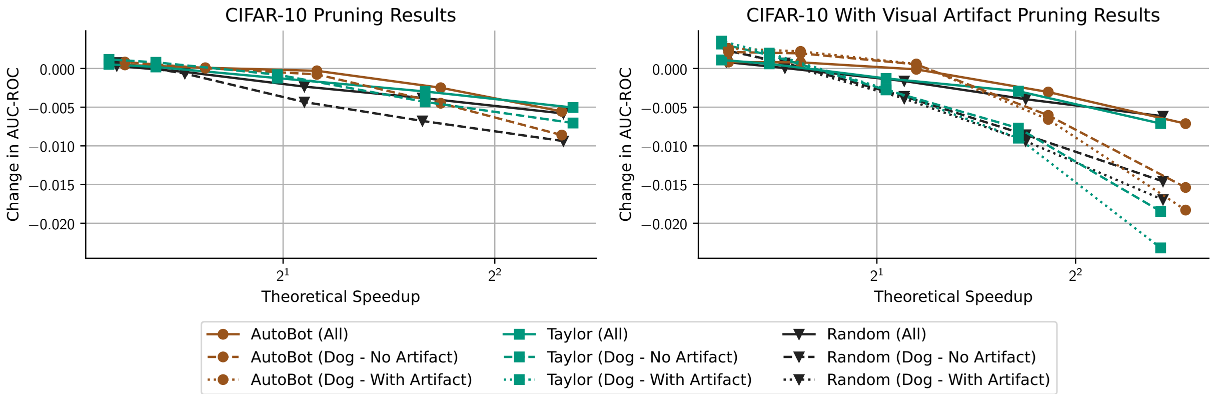


Figure 3.9: Mean pruning performance of the ResNet-18 and VGG-16 models with the CIFAR-10 dataset with visual artifact.

3.5 Summary

In this chapter, the effects of pruning on fairness was examined. Convolutional neural networks that were trained using the CelebA, Fitzpatrick17k and CIFAR-10 datasets were pruned using three structured pruning methods.

When networks trained to classify images from the CelebA dataset as blonde or non-blonde were pruned, the male subgroup saw a larger drop in ROC-AUC than the non-male subgroup. Adjusting the dataset to rectify disparities in label balance and/or gender balance reduced this difference. When networks trained to classify skin lesions using the Fitzpatrick17k dataset were pruned, existing biases were not aggravated. While the unpruned models were biased in favour of samples with a medium skin tone, the medium skin tone subgroup saw a greater decrease in ROC-AUC than the dark skin tone subgroup. Fairness issues during pruning were induced in CIFAR-10 models by manipulating the training data. Reducing the representation of two classes in the training data induced a bias against those classes during pruning for all pruning methods. Introducing a visual artifact that strongly correlated with the image labels induced a bias during pruning. However, the size of this effect varied across evaluated pruning methods.

These experiments demonstrate that pruning convolutional neural networks can, but not necessarily will, exacerbate fairness issues. Training set composition and image features that correlate with classification targets are two possible causes of this issue. The issue does not appear to be isolated to a particular pruning method. However, the effects experienced by each pruning method can vary in intensity. In the next chapter, a simple approach for improving the fairness of convolutional neural network pruning is explored.

Chapter 4

The Performance Weighted Loss Function

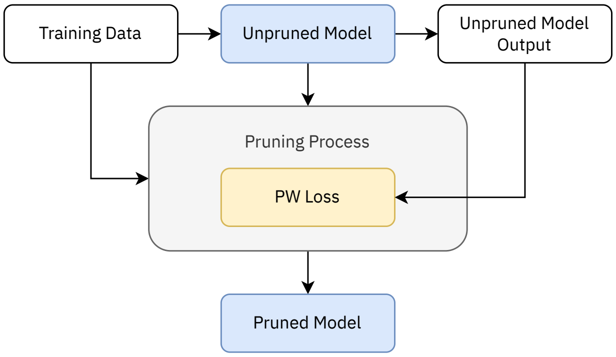


Figure 4.1: Overview of the performance weighted loss (PW Loss). The PW loss modifies the cross entropy loss of existing pruning methods to improve fairness through the use of the unpruned model output.

In Chapter 3 the fair pruning problem was analyzed. This analysis revealed that pruning can exacerbate existing model biases for some models. In this chapter we propose a simple method to improve the fairness of existing pruning methods. We call our method the performance weighted loss function.

We propose the performance weighted loss function as a simple method for boosting the fairness of data-driven methods for pruning convolutional filters in convolutional neural network image classifiers. As depicted in Figure 4.1, the performance weighted loss uses the output of the unpruned model to improve fairness during the pruning process. The loss function consists of two small tweaks to the standard cross-entropy loss function to prioritize the model's performance for poorly-classified samples over well-classified samples. These tweaks can be used to extend existing data-driven pruning methods without requiring explicit attribute information.

We demonstrate the effectiveness of our approach by pruning classifiers using two different pruning approaches for the CelebA [27], Fitzpatrick17k [10] and CIFAR-10 [24] datasets. Our results show that the performance weighted loss function can enable existing pruning methods to prune neural networks without significantly increasing model bias.

4.1 Motivation

In the fair pruning problem, model performance can be significantly impacted for certain sample subgroups. The highly impacted subgroups were characterized by poor representation in the training data or worse subgroup performance by the original model when compared to unimpacted groups. The performance decrement induced by the pruning process disproportionately impacts subgroups which are underrepresented and poorly classified.

To rectify this inequality, we can design a pruning process that prioritizes maintaining the performance of samples from potentially impacted subgroups. However, we do not need to develop a new pruning method from scratch to achieve this objective. Many existing pruning methods use data to identify which model parameters should be removed. Some methods use parameters learned via a loss minimization process whereas others values derived from gradients calculated with respect to a loss function. By modifying the loss function to prioritize samples from impacted subgroups, we can boost the fairness of existing pruning methods.

4.2 Method

We make two different modifications to the standard cross-entropy loss function to transform it into the performance weighted loss function (PW loss). We first apply sample weighting to ensure that samples from impacted groups have a larger contribution to the loss function. We then transform the sample labels to ensure that we are not reinforcing undesirable model behaviours.

As the attribute information required to identify impacted subgroups is not always readily accessible, our weighting scheme does not depend on any external information. We instead use the output of the original model to determine each sample weight. We assign larger weights to samples for which the original model was not able to confidently classify. The form of the scheme resembles the focal loss [26]. However, as the samples are weighted using the outputs of the original model the weights do not depend on the current output of the model and will not change during training. The weight assigned to the ith data sample, w_i, is given by the following equation:

$$w_i = \theta + (1 - \hat{y}_i)^\gamma \tag{4.1}$$

where $\hat{y}_i$ is the predicted probability given by the original model for the sample's true class, $\theta \in [0, 1]$ is the minimum weight value and $\gamma \geq 0$ controls the shape of the relation between $\hat{y}_i$ and w_i.

We also emphasize the model performance through the use of corrected soft-labels in the cross-entropy function. Rather than using the true labels of each sample, we use the output of the original model for the loss function in the pruning process. Without this change, the preservation of an originally poorly classified sample's prediction probability would result in a greater loss value than the preservation of an originally well classified sample's prediction probability. The use of true labels implicitly prioritizes the preservation of model performance for samples that have predictions closer to their true labels. Using the model output as soft-labels alleviates this implicit prioritization.

However, as we are assigning higher weights to samples that are originally classified by the original model while also using the original model's output as our labels, we are consequently assigning the highest weights to incorrect labels. To avoid emphasizing incorrect behaviours we correct the soft-labels. The corrected soft-label, $\hat{\boldsymbol{y}}_i^*$ is defined as:

$$\hat{\boldsymbol{y}}_i^* = \begin{cases} \hat{\boldsymbol{y}}_i & \text{if} \quad \hat{C}_i = C_i \\ \boldsymbol{y}_i & \text{otherwise} \end{cases} \tag{4.2}$$

where $\hat{\boldsymbol{y}}_i$ contains the prediction probabilities derived from the model output for the ith sample, $\boldsymbol{y}_i$ is the true label vector of the ith sample, $\hat{C}_i$ is predicted class of the ith sample and C_i is the true class of the ith sample. The corrected soft-label takes on the value of the model's prediction probabilities when the prediction is correct and the true label when the prediction is incorrect.

By the application of the performance weighted scheme and corrected soft-labels onto the standard cross-entropy function, the PW loss function, $\mathcal{L}_{PW}$, is defined by:

$$\mathcal{L}_{PW} = \sum_{i=1}^{N} w_i l_{CE}(\hat{\boldsymbol{y}}_i^*, \hat{\boldsymbol{y}}_i') \tag{4.3}$$

where $\hat{\boldsymbol{y}}_i'$ contains the prediction probabilities derived from the model output for the ith sample after pruning, $l_{CE}(\hat{\boldsymbol{y}}_i^*, \hat{\boldsymbol{y}}_i')$ is the cross-entropy between the corrected soft-label and the prediction probabilities of the pruned model for the ith sample, and N is the batch size.

By using this loss function with existing data-driven pruning methods, we can reduce the bias exaggerating effect of pruning by emphasizing samples that are more likely to be negatively affected by pruning.

4.3 Experiments

4.3.1 Experimental Set-up

We applied the PW loss to the same pruning methods used in Chapter 3: the 'AutoBot' method of Castells and Yeom [4] and the 'Taylor' method of Molchanov et al. [33]. For both methods, we pruned whole convolutional filters rather than individual neurons.

In the AutoBot method, the bottlenecks are optimized by minimizing a loss function that includes the cross-entropy between the original and pruned model outputs, as well as terms that encourage the bottlenecks to limit information moving through the model, achieving a target number of FLOPS [4]. We applied the performance weighted loss function to the method by replacing the cross-entropy term in the loss function with the performance weighted loss function. Additionally, we also used the performance weighted loss function when retraining the model after pruning.

The importance metric of the Taylor expansion method is formed using the gradient of the loss function with respect to each feature map and the value of each feature map [33]. This method alternates between training the network and pruning a filter. In our implementation, a filter is pruned every five iterations. We applied the performance weighted loss function by replacing the loss functions used in the gradient calculation and model training with the performance weighted loss function. Once again, we also used the performance weighted loss function when retraining the model after pruning.

For all methods we used the same hyperparameters that were used in Chapter 3. This means that hyperparameters for the pruning methods were selected without the PW loss applied and were used for both unmodified and PW loss method variants. We repeated each experiment three times. All figures displaying model performance after pruning are displaying the average of all trials. The figures also include the pruning results of the unmodified methods from Chapter 3.

As in Chapter 3, we compared the change in the areas under the receiver operator curves (ROC-AUC) for various subgroups for five different degrees of pruning. For non-binary classification we used the average one-vs-rest ROC-AUC. We once again measured the degree to which a model is pruned using the theoretical speedup.

4.3.2 Evaluating Fairness and Performance

After incorporating the PW loss into the AutoBot and Taylor pruning methods, we repeated some of our experiments from Chapter 3 to measure the impact of the PW loss on fairness during pruning. The CelebA face classification task, the Fitzpatrick17k skin lesion classification task, the CIFAR-10 classification task, the CIFAR-10 with underrepresentation task and the CIFAR-10 with visual artifact classification task were used. The models that were trained for 3 were once again pruned for these new experiments. Additionally, an EfficientNet-V2 Small [44] model was trained as a second CIFAR-10 model.

Pruning the CelebA Models

The ResNet-18 [14] model and VGG-16 [42] model trained for the CelebA task were pruned using target theoretical speedups of 16, 32, 64, 128 and 256.

The change in ROC-AUC for all tested pruning methods with and without the PW loss for the ResNet-18 and VGG-16 models can be found in Figure 4.2. Some of the VGG-16 models pruned using the AutoBot and AutoBot with performance weighting methods always predicted a single class. These degenerate models were not included in the mean values plotted in the graph.

Performance weighting improved the fairness of pruning the ResNet-18 model with the AutoBot and Taylor pruning methods, and the VGG-16 model with the Taylor pruning method. The introduction of performance weighting improved the post-pruning male ROC-AUCs for these methods. For all models and pruning methods, the PW loss appears to be more impactful for male samples than non-male samples. At low theoretical speedups, the pruning methods with performance weighting were able to compress the model with only small changes in fairness.

There were some situations in which performance weighting was not beneficial. At higher theoretical speedups with the VGG-16 model, the PW loss was not significantly beneficial. Additionally, applying performance weighting to the AutoBot method with the ResNet-18 model appeared to prevent the method from achieving higher target theoretical speedups.

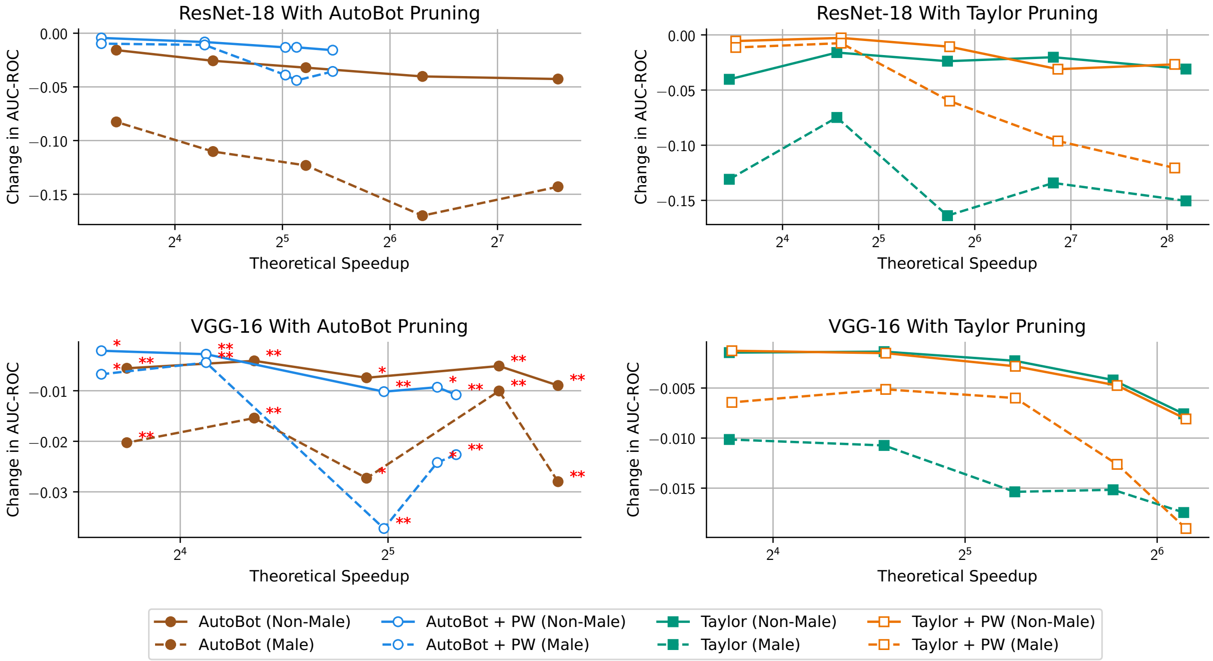


Figure 4.2: Mean pruning performance of the ResNet-18 and VGG-16 models with and without the performance weighted loss function on the CelebA task.

Pruning the Fitzpatrick17k Models

The ResNet-34 [14] model and a EfficientNet-V2 Medium [44] model for the Fitzpatrick17k task were pruned using target theoretical speedups of 2, 4, 8, 16 and 32.

While the unpruned models exhibited a moderate bias against darker skin tones, pruning methods with and without the PW loss did not exacerbate this bias. This can be seen in Figure 4.3. For all models, the performance weighted loss only had a small positive impact on performance at the lowest tested theoretical speedup. At other theoretical speedups it had a negative or neglible impact on performance for both medium and dark subgroups.

These results indicate that performance weighting is not an appropriate solution for all datasets and models that exhibit bias. Performance weighting is not a method for debiasing a model. It is intended to limit the fairness impact of pruning. It may not be beneficial in situations in which the impact of pruning does not exacerbate the exisitng biases of the model.

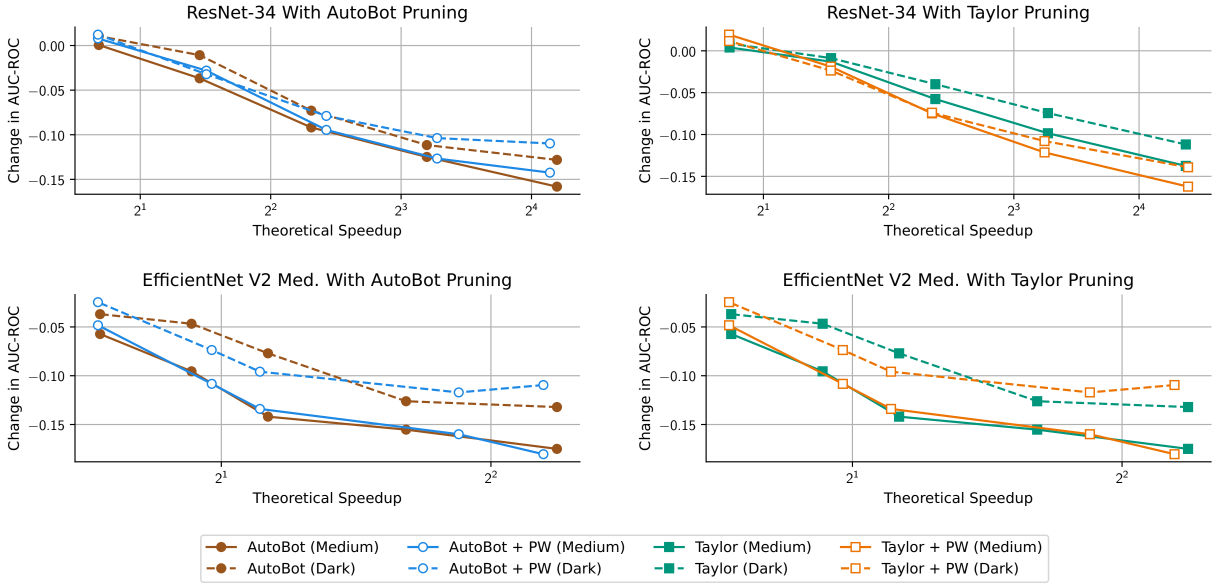


Figure 4.3: Mean pruning performance of the ResNet-34 and EfficientNet-V2 Med. models with and without the performance weighted loss function on the Fitzpatrick17k task.

Pruning the CIFAR-10 Models

The ResNet-56 [14] model and EfficientNet-V2 Small [44] models for the CIFAR-10 task were pruned using target theoretical speedups of 1.33, 2, 4, 8 and 16.

The ROC-AUCs of the ResNet and EfficientNet models were 0.9957 and 0.9995 respectively. These models have no known fairness issues. However, we still applied the PW loss to the pruning methods used with these models to understand its effect on the overall performance of models without known biases.

As shown in Figure 4.4, the PW loss had very different impact on the pruning processes for both models. For the ResNet-56 model, it improved performance for both tested pruning methods at all theoretical speedups. In contrast, with the EfficientNet-V2 Small model, it had a negligible impact for AutoBot pruning and a detrimental impact for Taylor pruning.

A pruning approach that prioritizes the performance of poorly classified samples can help improve the performance of models that do not have known fairness concerns. These models may benefit from the PW loss due to hidden biases that are aggravated by pruning. However, such an approach is not beneficial for all models. The PW loss should not be applied indiscriminately.

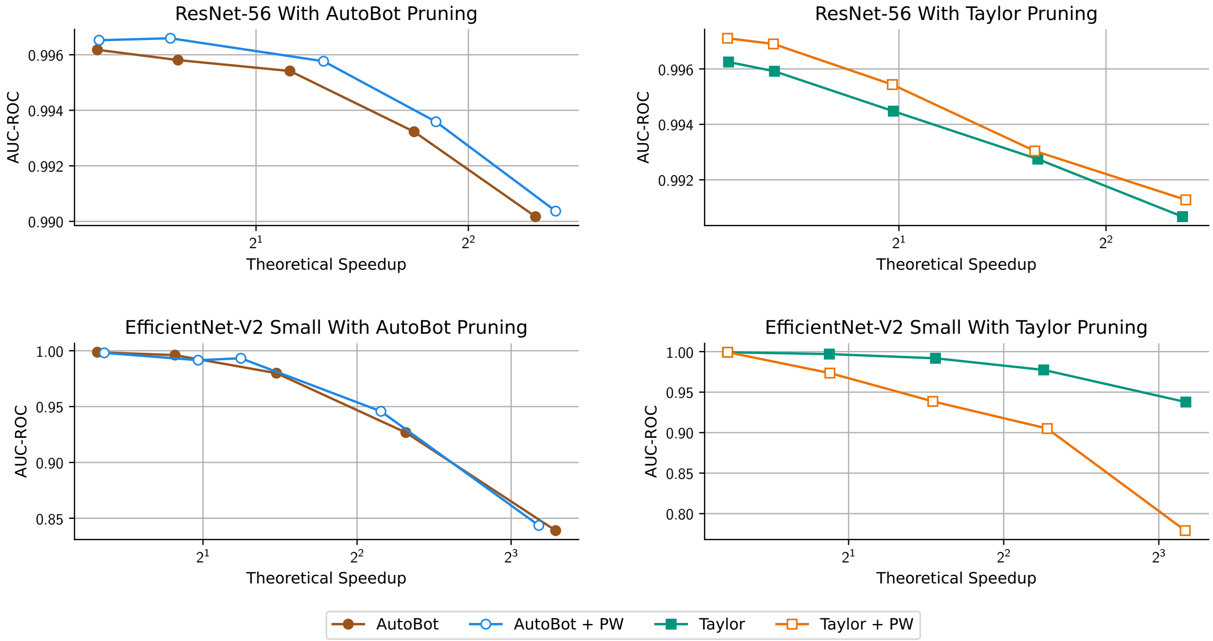


Figure 4.4: Mean pruning performance of the ResNet-56 and EfficientNet-V2 Small models with and without the performance weighted loss function on the CIFAR-10 task.

Pruning the CIFAR-10 Models With Underrepresentation and Visual Artifact

The ResNet-56 [14] model for the CIFAR-10 with underrepresentation task and the ResNet-56 [14] model for the CIFAR-10 with visual artifact task described in 3 were pruned using target theoretical speedups of 1.33, 2, 4, 8 and 16.

As depicted in Figure 4.5, all model-method combinations saw moderate improvements in fairness due to the introduction of performance weighting. Performance weighting increased the ROC-AUC of the affected classes more than the overall ROC-AUC. This improvement was smaller at higher theoretical speedups for the Taylor method while the impact was more consistent at all theoretical speedups for the AutoBot method.

Performance weighting was able to improve the fairness of pruning when the source of the fairness concern was underrepresentation and when the source of the fairness concern was a visual artifact that correlates with the class label. Performance weighting appears to be beneficial for models trained with a variety of bias sources.

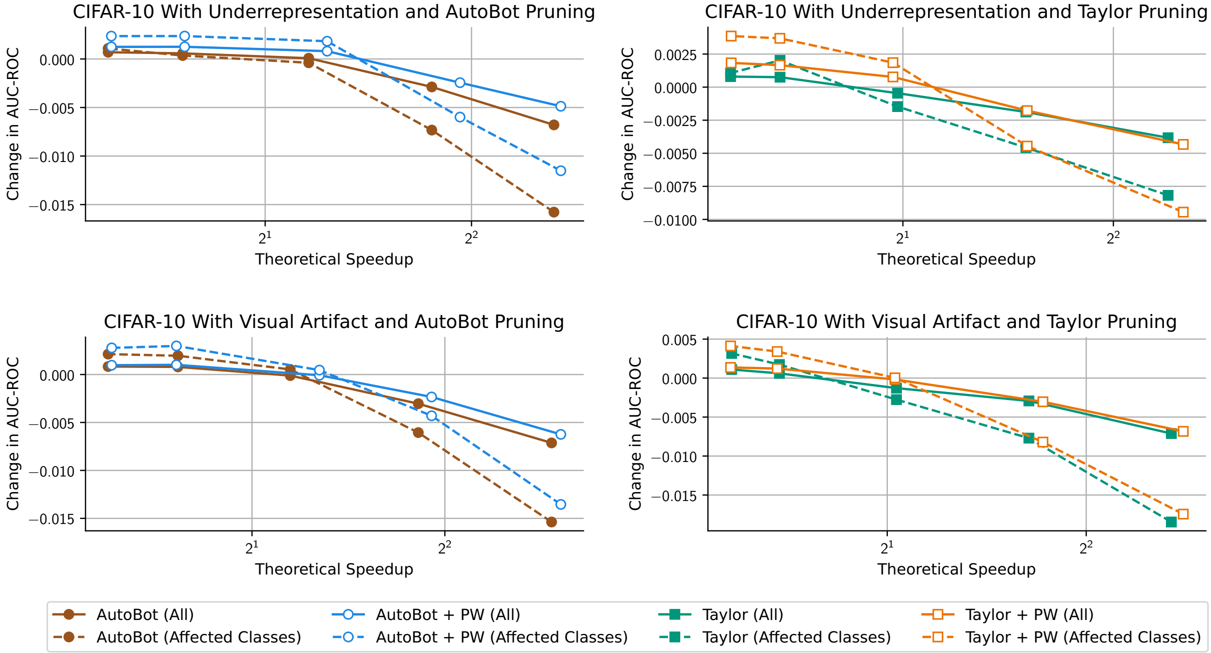


Figure 4.5: Mean pruning performance of the ResNet-56 model with and without the performance weighted loss function on the CIFAR-10 with underrepresentation and CIFAR-10 with visual artifact tasks.

4.3.3 Loss Analysis

To understand the impact the PW loss has during training, we analyzed the loss values of each batch during pruning. Figure 4.6 depicts the change in loss values for each attribute and class value for the first 200 batches of pruning the ResNet-18 CelebA model using the AutoBot and Taylor methods after the PW loss was applied. The models were pruned using a target theoretical speedup of 16 and the same random seed was used for batch selection for all depicted experiments. While the precise number of samples in each subgroup randomly

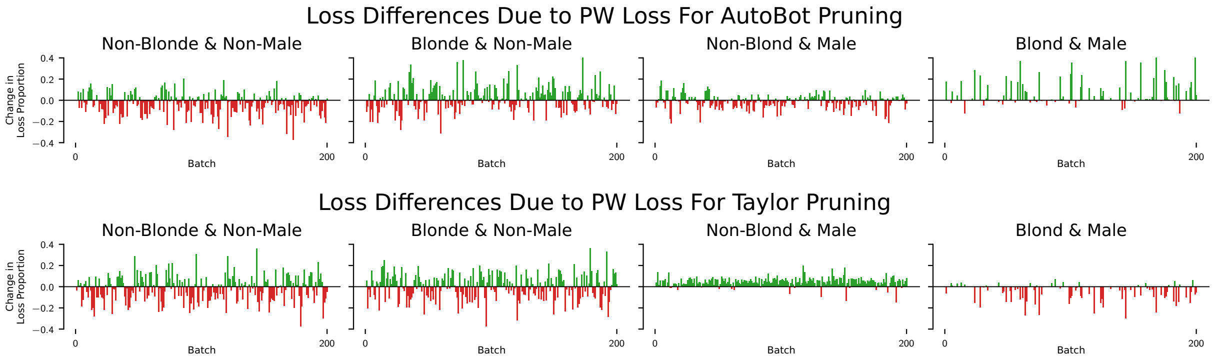


Figure 4.6: Change in the proportion of the batch loss due to the use of the PW loss, segmented by class and attribute.

vary between batches, the number of samples in each subgroup for the entire training set can be found in Table 3.1.

The introduction of the PW loss to the AutoBot pruning procedure causes a clear increase in loss contribution of the blond male samples. Blond male samples are poorly represented and poorly classified by the original model. An increase in the loss contribution demonstrates that the PW loss is causing the model pruning process to place more emphasis on the performance for this group to improve fairness. In contrast, the introduction of the PW loss to the Taylor pruning procedure instead increases loss contribution of the non-blond male samples and decreases the loss contribution of the blond male samples.

Despite the difference in loss contributions, in the results depicted in Figure 4.2, we did see improved fairness between male and non-male samples for both pruning methods at the utilized theoretical speedup with the use of the PW loss. This may be explained by the both methods seeing an increased loss contribution from male samples after the introduction of the PW loss, even if the specific samples which see an increase differ between pruning methods. While the PW loss did not appear to have the same effect on the two pruning methods, it was able to consistently increase the loss contribution of poorly represented samples without explicit attribute information.

4.3.4 Ablation

To measure the effects of the components of the PW loss independently, we pruned our ResNet-18 CelebA model using the AutoBot method with only the corrected soft-labels and with only the weighting scheme described in equation 4.1. We applied the modifications to the only the pruning process, to only the retraining process and to both the pruning and retraining processes.

The ablation results can be found in Figure 4.7. Both the modifications were only effective when the retraining process was modified, indicating that simply modifying the process by which parameters are selected to be pruning is insufficient to mitigate the effects of bias. The corrected soft labels were similarly beneficial when they were applied to pruning and retraining as they were when applied to just retraining. The weighting scheme was most beneficial when applied only to the retraining process. Different parts of the performance weighted loss appear to be more impactful for different parts of the pruning process. In some situations is may be beneficial to solely apply the components of the performance weighted loss. In particular, the use of corrected soft-labels does not require the selection of any parameters, allowing it to be used in situations in which parameter selection is not feasible.

We can also see that each test either has a similar effect as the full method or little effect at all. It is possible that there is a tipping point at which any applied fairness boosting methods cause the pruning process to converge to a fair solution instead of the original unbiased solution. If the process is already converging to the fair solution, any additional fairness boosting methods would have a small effect. If the fairness boosting methods are insufficient to encourage convergence to a fair solution, the effect of the methods would be minimal.

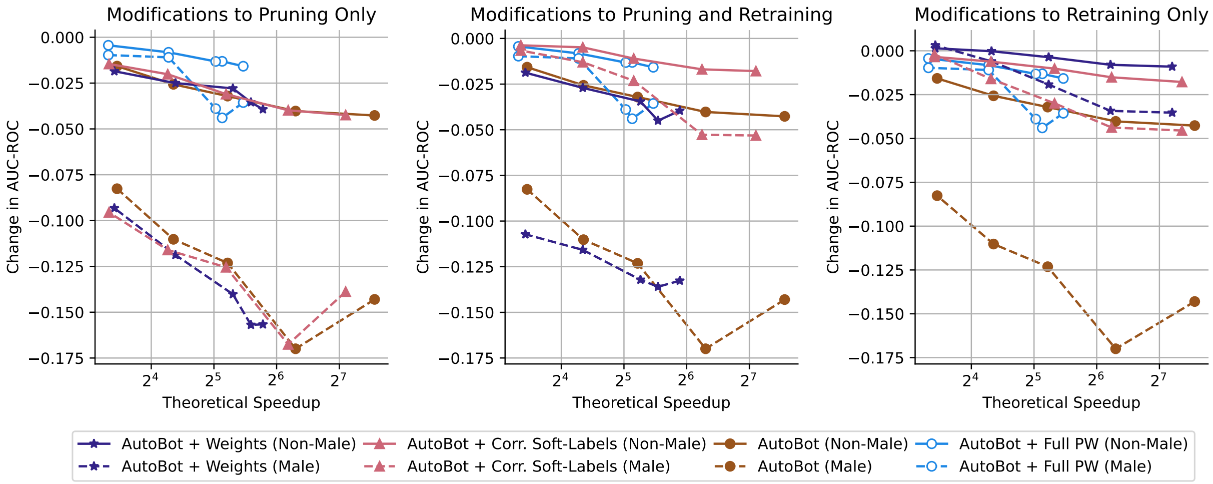


Figure 4.7: Pruning performance of the ResNet-18 models with the CelebA dataset when elements of PW loss are applied independently to the pruning process (left), and to the pruning process as well as the post-prune retraining process (right).

4.3.5 Summary

In this chapter, we presented the performance weighted loss function as a novel method for mitigating the impact of convolutional neural network pruning on fairness. The performance weighted loss function is a simple modification that can be applied to any pruning method that uses the cross-entropy loss. Our experimental results indicate that the performance weighted loss function can help prevent model biases from becoming exacerbated in many, but not all, pruning processes. The performance weighted loss function is a useful tool for practitioners who seek to compress existing models without introducing new fairness concerns.

Chapter 5

Conclusion

In this chapter, we summarize our key contributions detailed in Chapters 3 and 4, make recommendations, identify the limitations of our work, outline potential future work and discuss the implications of this thesis.

5.1 Summary of Contributions

In this thesis, the problem of fair pruning of convolutional neural networks was examined. In this problem, existing model biases are exacerbated by pruning for compression. Consequently, models that are pruned to be able to be utilized in resource constrained environments may be ill-suited for fairness sensitive environments. Previous work had identified this problem and established a relationship between fairness and model compression, but the problem was not well explored. To explore this problem, we analyzed the fairness effects of pruning in a variety of situations in Chapter 3 and proposed a simple modification to the loss function to promote fairness during pruning in Chapter 4.

In Chapter 3, the effects of pruning on fairness was examined in a variety of situations. Experiments were conducted using three different structured pruning methods and three different datasets. These experiments revealed that pruning can have negative effects on fairness. This effect does not appear to be guaranteed but it also does not appear to be limited to particular pruning methods. The effect was influenced by manipulating the datasets, identifying dataset composition and the presence of a visual feature that correlates with the classification target as potential causes for this effect.

In Chapter 4, the performance weighted loss function was proposed as a simple method for improving the fairness of existing data-driven pruning methods. The loss function is formed through two simple tweaks to the standard cross-entropy loss function that encourage the pruning process to prioritize maintaining the performance of poorly classified samples. By using sample weighting and corrected soft-labels, the performance weighted loss was able to improve the fairness of many of the tested classification tasks. While the loss function is not universally beneficial, it is a useful tool for machine learning practitioners seeking to prune networks in fairness sensitive environments.

5.2 Recommendations

The findings of this thesis lead us to make two recommendations: Machine learning practitioners should consider fairness when pruning networks and machine learning researchers should consider more than model performance when proposing new pruning methods.

Our experimental results demonstrate that pruning can negatively affect fairness and that simple changes to the pruning process, such as the use of the performance weighted loss, can mitigate this effect. Consequently, we believe it is important that practitioners applying model pruning to real-world systems consider the fairness impact of their actions. Practitioners should consider the social context of their system, identify the fairness issues that could arise from pruning, test the fairness of the system and explore modifications to the pruning process that could improve fairness. Practitioners should proceed thoughtfully rather than simply assuming that pruning will have a neutral effect on fairness.

Machine learning researchers developing new pruning methods can aid practitioners in the design of fair machine learning systems by describing the total impact of their pruning method instead of solely focusing on overall performance. Most pruning methods aim to preserve model performance. However, as we have demonstrated in this thesis, some subgroups may experience greater than average changes in performance due to pruning. Additional experiments to identify the properties of samples that are most negatively affected by a pruning method could help practitioners understand which subgroups will be most affected by the application of the pruning method.

5.3 Limitations

While we were able to demonstrate the fairness effects of pruning and the impact of the performance weighted loss in a variety of situations, we were only able to evaluate pruning with a small number of models and a small number of pruning methods. More experimentation would be needed to identify the the conditions of environments in which practitioners should be concerned about fairness during pruning. Similarly, evaluating the fairness impacts of additional pruning methods would provide a more complete understanding into how each pruning design decision affects fairnes. More work is required to understand how pruning affects fairness in specific situations and how to best address the fairness concerns of pruning for each situation.

5.4 Future Work

The fair pruning problem is neither solved nor fully understood. Additional work is required to enable practitioners to easily, reliably and confidently utilize model pruning without introducing any potential fairness concerns. Three key areas that could be addressed by future work are identifying the properties of pruning methods that affect fairness, exploring more sophisticated methods for improving model fairness and developing a fair pruning benchmark.

5.4.1 Identifying the Properties of Pruning Methods That Affect Fairness

Our results indicate that different models can respond differently to different pruning methods with respect to fairness and that simple tweaks to the pruning process can improve fairness. However, the causes of this effect are not clear. Developing an understanding of how the properties of the data, models and pruning methods all influence fairness during pruning would allow for more targeted and sophisticated methods for addressing the problem of fair pruning.

To develop this understanding, we will need to conduct more pruning experiments to evaluate the fairness of pruning in various conditions. However, rather than simply observing the fairness impact of pruning, we would need to influence the impact through modifications to the pruning method, data and model. For instance, we could examine the fairness of pruning on models trained using datasets with differing degrees of imbalances or examine how each specific pruning design decision affects pruning in various contexts.

5.4.2 Exploring More Sophisticated Methods for Improving Model Fairness

The development of more sophisticated methods of fair pruning would provide machine learning practitioners the tools they need to apply pruning to real-world models without introducing fairness concerns. The method proposed in this thesis, the performance weighted loss function, does appear to improve fairness in many contexts. However, it is a simple approach and it is not beneficial for every model. Exploring more sophisticated approaches could yield fair pruning methods that are more targeted, robust and adaptable.

5.4.3 Developing a Fair Pruning Benchmark

We believe that the development of a fair pruning benchmark could help accelerate the development of fair pruning methods. While there are datasets suitable for fairness analyses [10, 12] and common model-dataset combinations used to evaluate pruning methods [1], there is currently no standardized approach for measuring the fairness of a pruning method. Developing a fair pruning benchmark would allow researchers to easily compare the fairness impacts of their pruning methods with prior methods. This would likely involve the identification of the creation or identification of a dataset with attribute information, training a model with known fairness issues to be pruned, and the identification of an appropriate metric.

5.5 Discussion

As machine intelligence systems become increasingly sophisticated and integrated into our everyday lives, the importance of ensuring those systems have a positive social impact grows. We need to ensure that ML systems are not pushing human workers into poverty, spreading misinformation and perpetuating systemic discrimination.

These problems can not be solved by clever engineering alone. Our social, political and economic systems must adapt to new developments in ML. However, ML researchers and practitioners still have an important role to play. Technical developments are an essential part of the solution to the problem of ethical ML. Without understanding the abilities and limitations of ML systems, we can not predict the impact they will have on society. Without knowledge of methods for making ML systems more ethical, we have little recourse when ethical issues are found.

In this thesis, we furthered our understanding of how model pruning affects fairness and developed a simple method to address this issue. We hope that our work will help ensure the fairness of future machine learning systems that utilize model pruning.

References

[1] Davis Blalock et al. "What is the state of neural network pruning?" In: *Proceedings of machine learning and systems* 2 (2020), pp. 129–146.

[2] Daniel Borkan et al. "Nuanced Metrics for Measuring Unintended Bias with Real Data for Text Classification". In: *Companion Proceedings of The 2019 World Wide Web Conference*. WWW '19. New York, NY, USA: Association for Computing Machinery, May 2019, pp. 491–500. ISBN: 978-1-4503-6675-5. DOI: 10.1145/3308560.3317593.

[3] Kathleen Cachel and Elke Rundensteiner. "FINS Auditing Framework: Group Fairness for Subset Selections". In: *Proceedings of the 2022 AAAI/ACM Conference on AI, Ethics, and Society*. AIES '22. Oxford, United Kingdom: Association for Computing Machinery, 2022, pp. 144–155. ISBN: 9781450392471. DOI: 10.1145/3514094.3534160. URL: https://doi.org/10.1145/3514094.3534160.

[4] Thibault Castells and Seul-Ki Yeom. "Automatic neural network pruning that efficiently preserves the model accuracy". In: *arXiv preprint arXiv:2111.09635* (2021).

[5] Jeffery Dastin. "Amazon scraps secret AI recruiting tool that showed bias against women". en. In: *Reuters* (Oct. 2018). URL: https://www.reuters.com/article/us-amazon-com-jobs-automation-insight-idUSKCN1MK08G (visited on 04/16/2023).

[6] Mauricio Delbracio et al. "Mobile computational photography: A tour". In: *Annual Review of Vision Science* 7 (2021), pp. 571–604.

[7] Xiaohan Ding et al. "ResRep: Lossless CNN Pruning via Decoupling Remembering and Forgetting". In: *Proceedings of the IEEE/CVF International Conference on Computer Vision*. 2021, pp. 4510–4520.

[8] William Falcon and The PyTorch Lightning team. *PyTorch Lightning*. Mar. 2019. DOI: 10.5281/zenodo.3828935. URL: https://github.com/Lightning-AI/lightning.

[9] Gongfan Fang. *Torch-Pruning*. Version 0.2.7. July 2022. URL: https://github.com/VainF/Torch-Pruning.

[10] Matthew Groh et al. “Evaluating Deep Neural Networks Trained on Clinical Images in Dermatology with the Fitzpatrick 17k Dataset”. In: *arXiv preprint arXiv:2104.09957* (2021).

[11] Moritz Hardt, Eric Price, and Nati Srebro. “Equality of opportunity in supervised learning”. In: *Advances in neural information processing systems* 29 (2016).

[12] Caner Hazirbas et al. “Towards measuring fairness in ai: the casual conversations dataset”. In: *IEEE Transactions on Biometrics, Behavior, and Identity Science* 4.3 (2021), pp. 324–332.

[13] Kaiming He et al. “Deep residual learning for image recognition”. In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2016, pp. 770–778.

[14] Kaiming He et al. “Deep residual learning for image recognition”. In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2016, pp. 770–778.

[15] Yang He and Lingao Xiao. “Structured Pruning for Deep Convolutional Neural Networks: A survey”. In: *arXiv preprint arXiv:2303.00566* (2023).

[16] Sara Hooker et al. “Characterising bias in compressed models”. In: *arXiv preprint arXiv:2010.03058* (2020).

[17] The White House. *Blueprint for an AI Bill of Rights - OSTP*. en-US. Oct. 2022. URL: https://www.whitehouse.gov/ostp/ai-bill-of-rights/.

[18] Andrew G Howard et al. “Mobilenets: Efficient convolutional neural networks for mobile vision applications”. In: *arXiv preprint arXiv:1704.04861* (2017).

[19] Yerlan Idelbayev. *Proper ResNet Implementation for CIFAR10/CIFAR100 in PyTorch*. https://github.com/akamaster/pytorch_resnet_cifar10. Accessed: 2023-01-24.

[20] Bhanu Jain, Manfred Huber, and Ramez Elmasri. “Increasing Fairness in Predictions Using Bias Parity Score Based Loss Function Regularization”. In: *arXiv preprint arXiv:2111.03638* (2021).

[21] Vinu Joseph et al. “Going Beyond Classification Accuracy Metrics in Model Compression”. In: (Dec. 2020). DOI: 10.48550/arXiv.2012.01604.

[22] Nikola Jovanović et al. “FARE: Provably Fair Representation Learning”. In: *arXiv preprint arXiv:2210.07213* (2022).

[23] Eungyeup Kim, Jihyeon Lee, and Jaegul Choo. "Biaswap: Removing dataset bias with bias-tailored swapping augmentation". In: *Proceedings of the IEEE/CVF International Conference on Computer Vision*. 2021, pp. 14992–15001.

[24] Alex Krizhevsky, Geoffrey Hinton, et al. "Learning multiple layers of features from tiny images". In: (2009).

[25] Zewen Li et al. "A Survey of Convolutional Neural Networks: Analysis, Applications, and Prospects". In: *IEEE Transactions on Neural Networks and Learning Systems* 33.12 (2022), pp. 6999–7019. DOI: 10.1109/TNNLS.2021.3084827.

[26] Tsung-Yi Lin et al. "Focal Loss for Dense Object Detection". In: *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*. Oct. 2017.

[27] Ziwei Liu et al. "Deep Learning Face Attributes in the Wild". In: *Proceedings of International Conference on Computer Vision (ICCV)*. Dec. 2015.

[28] Ilya Loshchilov and Frank Hutter. "Decoupled weight decay regularization". In: *arXiv preprint arXiv:1711.05101* (2017).

[29] Ilya Loshchilov and Frank Hutter. "Sgdr: Stochastic gradient descent with warm restarts". In: *arXiv preprint arXiv:1608.03983* (2016).

[30] Ričards Marcinkevičs, Ece Ozkan, and Julia E Vogt. "Debiasing Deep Chest X-Ray Classifiers using Intra-and Post-processing Methods". In: *arXiv preprint arXiv:2208.00781* (2022).

[31] Rachel Metz. *Twitter says its image-cropping algorithm was biased, so it's ditching it — CNN Business*. en. May 2021. URL: https://www.cnn.com/2021/05/19/tech/twitter-image-cropping-algorithm-bias/index.html.

[32] Shervin Minaee et al. "Image Segmentation Using Deep Learning: A Survey". In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 44.7 (2022), pp. 3523–3542. DOI: 10.1109/TPAMI.2021.3059968.

[33] Pavlo Molchanov et al. "Pruning convolutional neural networks for resource efficient inference". In: *arXiv preprint arXiv:1611.06440* (2016).

[34] James O' Neill. "An overview of neural network compression". In: *arXiv preprint arXiv:2006.03669* (2020).

[35] Thomas Norrenbrock, Marco Rudolph, and Bodo Rosenhahn. "Take 5: Interpretable Image Classification with a Handful of Features". In: *arXiv preprint arXiv:2303.13166* (2023).

[36] Michela Paganini. "Prune responsibly". In: *arXiv preprint arXiv:2009.09936* (2020).

[37] Adam Paszke et al. “PyTorch: An Imperative Style, High-Performance Deep Learning Library”. In: *Advances in Neural Information Processing Systems 32*. Ed. by H. Wallach et al. Curran Associates, Inc., 2019, pp. 8024–8035. URL: http://papers.neurips.cc/paper/9015-pytorch-an-imperative-style-high-performance-deep-learning-library.pdf.

[38] Inioluwa Deborah Raji et al. “Closing the AI accountability gap: Defining an end-to-end framework for internal algorithmic auditing”. In: *Proceedings of the 2020 conference on fairness, accountability, and transparency*. 2020, pp. 33–44.

[39] Shahar Segal et al. “Fairness in the eyes of the data: Certifying machine-learning models”. In: *Proceedings of the 2021 AAAI/ACM Conference on AI, Ethics, and Society*. 2021, pp. 926–935.

[40] Laleh Seyyed-Kalantari et al. “CheXclusion: Fairness gaps in deep chest X-ray classifiers”. In: *BIOCOMPUTING 2021: proceedings of the Pacific symposium*. World Scientific. 2020, pp. 232–243.

[41] Dinggang Shen, Guorong Wu, and Heung-Il Suk. “Deep Learning in Medical Image Analysis”. In: *Annual Review of Biomedical Engineering* 19.1 (2017). PMID: 28301734, pp. 221–248. DOI: 10.1146/annurev-bioeng-071516-044442. eprint: https://doi.org/10.1146/annurev-bioeng-071516-044442. URL: https://doi.org/10.1146/annurev-bioeng-071516-044442.

[42] Karen Simonyan and Andrew Zisserman. “Very deep convolutional networks for large-scale image recognition”. In: *arXiv preprint arXiv:1409.1556* (2014).

[43] Jacob Snow. *Amazon’s Face Recognition Falsely Matched 28 Members of Congress With Mugshots*. American Civil Liberties Union. July 26, 2018. URL: https://www.aclu.org/blog/privacy-technology/surveillance-technologies/amazons-face-recognition-falsely-matched-28 (visited on 04/16/2023).

[44] Mingxing Tan and Quoc Le. “Efficientnetv2: Smaller models and faster training”. In: *International Conference on Machine Learning*. PMLR. 2021, pp. 10096–10106.

[45] Wei Wang et al. “Development of convolutional neural network and its application in image classification: a survey”. In: *Optical Engineering* 58.4 (2019), p. 040901. DOI: 10.1117/1.OE.58.4.040901. URL: https://doi.org/10.1117/1.OE.58.4.040901.

[46] Yawen Wu et al. “Fairprune: Achieving fairness through pruning for dermatological disease diagnosis”. In: *Medical Image Computing and Computer Assisted Intervention–MICCAI 2022: 25th International Conference, Singapore, September 18–22, 2022, Proceedings, Part I*. Springer. 2022, pp. 743–753.

[47] Guangxuan Xu and Qingyuan Hu. “Can model compression improve nlp fairness”. In: *arXiv preprint arXiv:2201.08542* (2022).

[48] Jing Yang et al. “Using deep learning to detect defects in manufacturing: a comprehensive survey and current challenges”. In: *Materials* 13.24 (2020), p. 5755.

[49] Tao Zhang et al. “Fairness in Semi-Supervised Learning: Unlabeled Data Help to Reduce Discrimination”. In: *IEEE Transactions on Knowledge and Data Engineering* 34.4 (2022), pp. 1763–1774. DOI: 10.1109/TKDE.2020.3002567.

[50] Michael Zhu and Suyog Gupta. “To prune, or not to prune: exploring the efficacy of pruning for model compression”. In: *arXiv preprint arXiv:1710.01878* (2017).

[51] Zhuangwei Zhuang et al. “Discrimination-aware Channel Pruning for Deep Neural Networks”. In: *Advances in Neural Information Processing Systems*. Ed. by S. Bengio et al. Vol. 31. Curran Associates, Inc., 2018. URL: https://proceedings.neurips.cc/paper_files/paper/2018/file/55a7cf9c71f1c9c495413f934dd1a158-Paper.pdf.

[52] Zhengxia Zou et al. “Object Detection in 20 Years: A Survey”. In: *Proceedings of the IEEE* 111.3 (2023), pp. 257–276. DOI: 10.1109/JPROC.2023.3238524.

APPENDICES

Appendix A

Model Training and Pruning Parameters

To ensure transparency and enable reproducability, all parameters and procedures used to train, prune and retrain the models can be found below. All experiments were implemented using *PyTorch* 1.12.1 and *torchvision* 0.13.1 [37]. *PyTorch Lightning* 1.7.1 [8] was also used to train the models.

The ResNet-18 [14] CelebA model was trained for 20 epochs using the AdamW [28] optimizer with an initial learning rate of 0.0001 and a CosineAnnealingLR learning rate scheduler with $T_{max} = 20$ [29]. A batch size of 256 was used. The model was initialized using the provided ImageNet weights from *torchvision*. All parameters in layers except the final fully connected layer were frozen for the first 5 epochs after which they were unfrozen with a learning rate equal to 0.01 times the global learning rate. Early stopping was applied such that the parameters that achieved the lowest validation loss were saved after training.

The VGG-16 [42] CelebA model was trained for 10 epochs using the AdamW [28] optimizer with an initial learning rate of 0.0005 and a CosineAnnealingLR learning rate scheduler with $T_{max} = 10$ [29]. A batch size of 64 was used. The model was initialized using the provided ImageNet weights from *torchvision*. All parameters in layers except the final fully connected layer were optimized with a learning rate equal to 0.01 times the global learning rate. Early stopping was applied such that the parameters that achieved the lowest validation loss were saved after training.

The ResNet-34 [14] Fitzpatrick17k model was trained for 30 epochs using the AdamW [28] optimizer with an initial learning rate of 0.001 and a CosineAnnealingLR learning rate scheduler with $T_{max} = 30$ [29]. A batch size of 64 was used. The model was initialized using the provided ImageNet weights from *torchvision*. All parameters in layers except the final fully connected layer were frozen for the first 5 epochs after which they were unfrozen with a learning rate equal to 0.001 times the global learning rate.

The EfficientNet V2 Medium [44] Fitzpatrick17k model was trained for 30 epochs using the AdamW [28] optimizer with an initial learning rate of 0.001 and a CosineAnnealingLR learning rate scheduler with $T_{max} = 30$ [29]. A batch size of 32 was used. The model was initialized using the provided ImageNet weights from *torchvision*. All parameters in layers except the final fully connected layer were frozen for the first 5 epochs after which they were unfrozen with a learning rate equal to 0.01 times the global learning rate.

The ResNet-56 [14] CIFAR-10 model was built using the implementation from Idelbayev [19]. The model was trained for 200 epochs using the AdamW [28] optimizer with an initial learning rate of 0.1 for the final fully connected layer and 0.01 for all other parameters. The learning rate was multiplied by a factor of 0.1 at epochs 100, 150 and 175. A batch size of 256 was used.

The EfficientNet V2 Small [44] CIFAR-10 model was trained for 10 epochs with an initial learning rate of 0.001 and a CosineAnnealingLR learning rate scheduler with $T_{max} = 10$ [29]. A batch size of 128 was used. The model was initialized using the provided ImageNet weights from *torchvision*. All parameters in layers except the final fully connected layer were frozen for the first 5 epochs after which they were unfrozen with a learning rate equal to 0.01 times the global learning rate.

The parameter values used for our AutoBot [4] implementation can be found in Table A.1. β_{AB} refers to the parameter used by the AutoBot method to control the balance between the different terms of its loss function.

The parameter values used for our Taylor [33] implementation can be found in Table A.2. f_{prune} refers to the frequency of the pruning. That is, the number of batch iterations between the pruning of filters. N_{filters} refers to the number of convolutional filters that are pruned in each pruning instance.

The parameter values that are used for our PW losses can be found in Table A.3. Other parameters were not changed when the PW loss was introduced.

After pruning, all models were retrained using the AdamW [28] optimizer and CosineAnnealingLR learning rate scheduler with a T_{max} value equal to the number of epochs. The parameter values used to retrain the models can be found in Table A.4.

Table A.1: Parameters used for AutoBot pruning method

Dataset	Model	Learning Rate	Batch Size	Iters.	β_{AB}
CelebA	ResNet-18	0.85	64	200	2.7
CelebA	VGG-16	1.81	64	250	3.07
Fitzpatrick17k	ResNet-34	1.5	32	400	0.5
Fitzpatrick17k	EfficientNet V2 Med.	1.5	16	600	6.76
CIFAR-10	ResNet-56	0.7	128	200	5.3
CIFAR-10	EfficientNet V2 Small	1.88	64	200	2.0

Table A.2: Parameters used for Taylor pruning method

Dataset	Model	Learning Rate	Batch Size	f_{prune}	N_{filters}
CelebA	ResNet-18	0.01	64	5	1
CelebA	VGG-16	0.01	64	5	1
Fitzpatrick17k	ResNet-34	0.01	32	5	1
Fitzpatrick17k	EfficientNet V2 Med.	0.01	16	4	8
CIFAR-10	ResNet-56	0.01	128	5	1
CIFAR-10	EfficientNet V2 Small	0.01	64	5	8

Table A.3: Parameters used for PW loss

Dataset	Model	Base Method	θ	γ
CelebA	ResNet-18	AutoBot	0.3	1
CelebA	ResNet-18	Taylor	0.8	0.5
CelebA	VGG-16	AutoBot	0.75	3
CelebA	VGG-16	Taylor	0.9	5
Fitzpatrick17k	ResNet-34	AutoBot	0.8	2.5
Fitzpatrick17k	ResNet-34	Taylor	0.95	3
Fitzpatrick17k	EfficientNet V2 Med.	AutoBot	0.8	2
Fitzpatrick17k	EfficientNet V2 Med.	Taylor	0.95	3
CIFAR-10	ResNet-56	AutoBot	0.5	1
CIFAR-10	ResNet-56	Taylor	0.5	1
CIFAR-10	EfficientNet V2 Small	AutoBot	0.7	0.5
CIFAR-10	EfficientNet V2 Small	Taylor	0.7	0.5

Table A.4: Parameters used to retrain models

Dataset	Model	Learning Rate	Batch Size	Duration
CelebA	ResNet-18	0.0001	256	30 epochs
CelebA	VGG-16	0.0005	64	10 epochs
Fitzpatrick17k	ResNet-34	0.0001	64	30 epochs
Fitzpatrick17k	EfficientNet V2 Med.	0.00001	32	50 epochs
CIFAR-10	ResNet-56	0.001	256	200 epochs
CIFAR-10	EfficientNet V2 Small	0.00001	128	10 epochs