Imperial College
London

Final Report

Department of Computing

Imperial College of Science, Technology and Medicine

Counterfactual Data Augmentation for Deep Learning Predictive Models

Supervisor:
Prof. Ben Glocker

Co-supervisors:
Dr Miguel Monteiro
Dr Fabio De Sousa Ribeiro
and Dr Tian Xia

Author:
Izabella Kacprzak

Second Marker:
Dr Bernhard Kainz

June 19, 2023

Abstract

Fidelity of predictive models is often limited when they rely overly on biases in training data. These biases occur due to many constraints in the processes employed for data generation. While there exist many methods intended to target data imbalance, they are often incapable of removing spurious correlations between attribute subgroups.

Counterfactual inference allows for the synthesis of plausible images from original samples by modifying certain attributes while keeping others intact, thus eliminating false associations. We describe a novel method of counterfactual image data augmentation as a debiasing technique for predictive models. Our method is capable of synthesizing new data for the purpose of boosting performance when it is limited by data scarcity, data attributes correlation and other types of bias. To that end two approaches are used: (i) dataset expansion via counterfactual data augmentation; (ii) modification of the training objective with a counterfactual regularisation term. Furthermore, we perform a comparison of the counterfactual method and some chosen commonly used debiasing techniques. The evaluation is focused both on fine-grained local performance in each predefined attribute subgroup and more general global performance.

Moreover, we perform bias evaluation, which allows us to test the ability of trained image classifiers to adapt to counterfactual examples and estimate their fairness against chosen attributes.

Acknowledgements

I would like to thank Prof. Ben Glocker, Dr Miguel Monteiro, Dr Fabio De Sousa Ribeiro and Dr Tian Xia for their invaluable support, insight and guidance during the supervision of this project.

I would also like to thank my personal tutor, Prof. Sophia Drossopoulou, for her support during the last four years.

Finally, I would like to express my gratitude to my friends and family, but especially my parents, Marysia and Arek, and my sister, Martyna, for always pushing me out of my comfort zone and supporting me throughout my degree.

Contents

Chapter 1

Introduction

1.1 Motivation

Deep neural networks quickly gain popularity as they find use cases in many areas of life. They are often used to substitute a tedious decision process or to identify patterns or correlations unnoticeable by human observers [1]. One field where the use of neural networks – more specifically convolutional neural networks – has been increasing in recent years is medicine, wherein machine learning (ML) systems are being widely used for radiology, pathology and dermatology [2, 3, 4].

However, using machine learning systems in safety-critical settings does not come without risks. One such risk is the difficulty in generalisation of an ML model. Training on an imbalanced dataset often leads to a biased model. This means that machine learning models are often found to be over-reliant on the correlations between labels and peripheral attributes, i.e. attributes which do not directly imply the label of a datapoint [5]. One often used visual example is a classifier trained to distinguish between cows and camels [6]. In the training data, we are most likely to find that the majority of camel pictures will have a desert background while cows will probably be surrounded by grass. This correlation between label and background leads to bias in trained models. Given an image of a camel on grass the model is likely to label it as a cow.

Currently there are many methods used to mitigate dataset imbalance. The most commonly employed include: (i) resampling (oversampling or undersampling); (ii) re-weighing the loss function; (iii) data augmentation methods such as flipping, rotations or blurring, to name a few. Data augmentations can also be tailored to given datasets, targeting specific machine learning tasks by introducing domain knowledge into the training process.

All of the aforementioned methods work to an extent and it is part of this project to study what their limitations are. That having been said, it is intuitively clear that they all come with certain shortcomings. It is often not clear why applying specific actions may improve the performance of a model. For example, it might be counter-intuitive to include blurred versions of medical images in the training dataset as by doing so we are implicitly assuming that blur may occur naturally in held-out data. Certain data augmentations might also introduce a change in the joint distribution of inputs and outputs, known as distribution shift [7]. Furthermore, in certain situations it is highly impractical to condition the model in such a way that would efficiently target the bias. Coming back to the camels and cows example, it would be very difficult to find images of cows in deserts, generate them using standard augmentation methods or even apply any model-driven debiasing methods effectively.

This leads us to the concept of counterfactuals and counterfactual data augmentation. Recent advances in generative neural networks have allowed for the generation of visually plausible image counterfactuals. Counterfactual images are synthetically generated versions of original, factual examples in counter-to-fact scenarios. They can be seen as answers to "what if"-type questions. Using causal interventions we can manipulate specific attributes and synthesize new counterfactual images based on existing data.

One particularly useful application of counterfactuals is for debiasing datasets used for machine learning tasks such as classification or regression by e.g. addressing dataset imbalance in a principled manner. By constructing realistic structural causal models of datasets, and with sufficient observational data, we can train deep generative neural networks to encode image features of interest which respect the causal relationships in the associated graph. We can then use the trained models to apply interventions on existing biased data and generate additional, counterfactual, datapoints. Those synthetic images can then be used to expand datasets used to train predictive models or even utilised in model-based debiasing approaches.

1.2 Contributions

The goal of this project is to determine to what extent image counterfactuals are an effective method for targeting bias and improving performance of models in imbalanced datasets.

As part of the project, we develop a comparison of this novel method with widely-used, well-established methods and build an evaluation pipeline to analyse the effects of each technique in a systematic manner.

We test the effectiveness of counterfactual data augmentation on synthetic datasets as well as a real-world dataset of medical images. For the former, we are in full control of the attributes and the relationships between them are known apriori. That is, we have access to the true causal data generating process. On the other hand, the real-world medical imaging dataset is much more complex as each datapoint includes multiple attributes pertaining to the patient. The data is highly imbalanced both in terms of the classes as well as the attributes and exhibits internal attribute correlations.

Furthermore, using counterfactual images, we perform bias evaluation via fairness analysis on the chosen setups.

We believe that the chosen setups enable a thorough evaluation of the counterfactuals data augmentation method in both real and synthetic settings.

Chapter 2

Background

In this chapter, we:

- provide an overview of image classification and the problems this sub field is generally designed to solve, as well as introduce convolutional neural networks and describe some commonly used architectures (Sections 2.1, 2.2),
- discuss different types of bias such as data scarcity or data mismatch and go into details regarding what the causal factors normally are and how each type of bias manifests itself in the dataset as well as in a trained predictive model (Section 2.3),
- introduce some widely-used, standard debiasing methods for mitigating dataset bias and discuss their potential flaws and shortcomings providing motivation for novel bias mitigation methods (Section 2.4),
- overview the necessary background on counterfactual inference, covering: (i) causality; (ii) deep generative models, and (iii) common architectures including the Deep Structural Causal Model (DSCM) architecture used in this project (Section 2.5),
- discuss some evaluation methods frequently used in similar works (Section 2.6),
- provide a review of relevant literature, including several works on counterfactual bias mitigation and model fairness evaluation (Section 2.7).

2.1 Image Classification

Image classification or image label prediction is a supervised learning problem where, given a training set of images and target labels pertaining to each images, the task is to train a model to recognise similar images to those seen at training time and assign correct labels to them. Such models are called image classifiers.

An image classifier is a function that for each image datapoint from the image space

X assigns a label in the predefined label space Y. Then given a classifier θ with a loss function $l : \theta \times (X \times Y) \to \mathbb{R}$ the usual approach is to find a classifier that minimises the expected loss l via empirical risk minimisation. However, if the observed training data is not sufficiently homogeneous, then focusing on average performance can lead to important subgroups being affected. For example, if we find that the dataset used to train the classifier is biased in any way and fails to accurately represent the real-world distribution, average performance can be especially misleading.

2.2 Convolutional Neural Networks

Convolutional Neural Networks (CNNs) have become the predominantly used type of model for most computer vision tasks. Typically, a CNN is composed of multiple elements such as convolution layers, pooling layers, and fully-connected layers. The learning process is based on backpropagation through those layers [1].

Convolution layers perform feature extraction, applying linear and non-linear operations via activation functions. They usually use kernels which are arrays of numbers and apply them to the input pixel values in a sliding-window fashion to detect features and generate feature maps of the input.

Pooling layers also perform feature extraction but typically place greater emphasis on high activations. For example, when max-pooling is applied the size of the feature map is downsampled by extracting the maximum values across local regions in the input.

In CNNs, fully-connected layers are used at the end to map the extracted features to class labels, making a prediction. They typically receive flattened feature maps and the final fully-connected layer will have the same number of neurons as the number of classes we are trying to classify.

2.2.1 ResNet

A Residual Network (ResNet) [8] is a convolutional neural network specifically designed to scale to thousands of convolutional layers. The general belief is that the deeper the neural network the better, however, there exists a problem with adding too many layers called *vanishing gradient* [9]. As the gradient is calculated and propagated through the network by repeated multiplication, its value vanishes quickly, approaching zero. It is also possible for the opposite to happen. When the gradients are large their multiplication also becomes large overtime – a problem called *exploding gradients*.

To solve this problem, ResNet introduces the “identity shortcut connection” which allows for skipping one or more layers. A typical ResNet block consists of convolution layers, batch norms and ReLU activations [10]. At the end of each block, the output is added to the original input (Figure 2.1). There are many ResNet flavours, such as ResNeXt [11] or Wide ResNet [12], but they all build on top of the basic block.

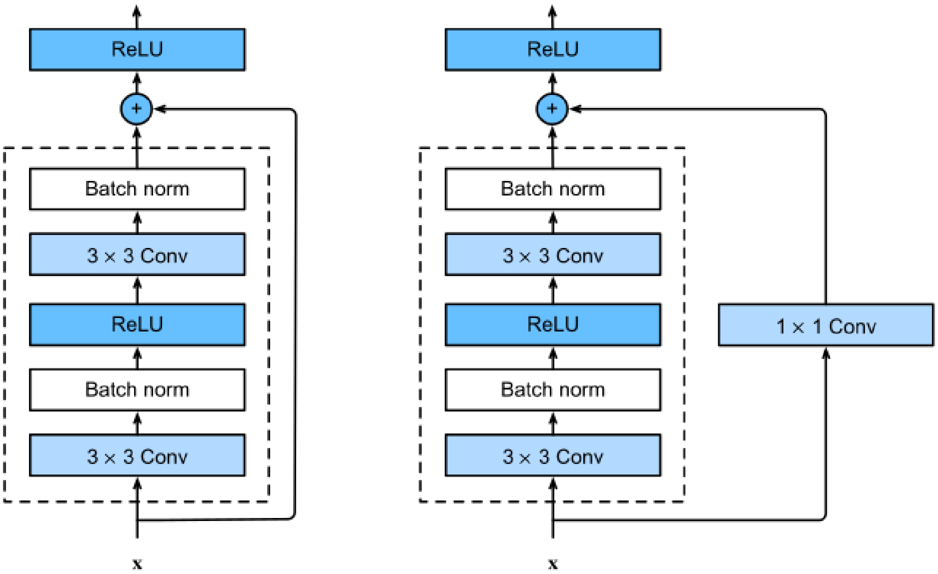

Figure 2.1: ResNet block with and without 1×1 convolution. Taken from [10]

2.2.2 DenseNet

DenseNet [13] is another convolutional neural network and it is often called an extension of ResNet. The main difference between DenseNet and ResNet is that DenseNet uses concatenation rather than addition to combine the inputs and outputs of a block. As a result, each layer is connected to every previous layer which creates a very dense dependency graph (hence DenseNet). DenseNet is built of Dense Blocks, each consisting of multiple convolution blocks that follow a similar structure to that of ResNet - convolution layer and batch normalization followed by an activation function. The outputs of all convolution blocks are then concatenated.

2.3 Dataset Bias

We talk about dataset bias when the distribution of datapoints in the training set is different from the distribution of real-world data. In a biased dataset, samples exhibit attributes which are not innate but rather correlated with target labels (i.e. bias attributes). For example, let us imagine a training dataset of dermatoscopic images on which we want to train a classifier to distinguish between a number of different types of skin lesions. The images in the dataset come from multiple sources and were taken using instruments of varying parameters, hence images produced by two different instruments have visible differences such as color resolution. A model trained on such a biased dataset will be likely to learn the bias attributes (e.g. the differences caused by varying resolution) rather than the intrinsic attributes that truly differentiate the various types of skin lesions. This means the model will perform poorly when tested on a dataset representing the attribute distribution as it appears in the real world.

There are many examples of such bias found in image classifiers [14, 15, 16, 17]. One such example is the use of imbalanced datasets for training gender classification systems. The authors in [14] aimed to evaluate three commercial gender classification systems on a balanced dataset they constructed which contained images of both male and female individuals of four different skin tones. They showed error rates of up to 34.7% on one of the classes (darker-skinned females). While the maximum error rate for lighter-skinned males was found to be 0.8%. They found that the classification systems have been trained on biased datasets which favoured lighter-skinned and male individuals which caused the bias in the final model. Systems like this one are often used to build facial analysis algorithms and studies like this show just how much attention they require.

Dataset bias can have many different causes. When dividing bias by its cause we can distinguish two main types: (i) data scarcity bias, and (ii) data mismatch bias.

2.3.1 Data Scarcity

It is very common – especially in medical contexts [18, 19, 20] – to encounter the problem of data scarcity. It is most often caused by high costs of data generation or hiring experts to annotate data points. Some of the techniques commonly used to fight this problem include semi-supervised learning and data augmentations. However, for semi-supervised learning to work, certain requirements must be met. Data points which belong to one class need to be similar to each other in terms of the attributes we perform the clustering on, since they will naturally get clustered together. Data augmentations often need to be tailored to specific datasets, for example it does not make much sense to flip MNIST images [21] as they do not represent the same digits anymore. Therefore, a certain amount of data analysis needs to be done before debiasing. Furthermore, many methods used are unintuitive and hard to reason about. For example, in the article introducing cutout [22] as a data augmentation method, the authors admit that they found the less intuitive, more random, version of cutout to work better and give superior results. However, they do not provide any reasoning for why that might be the case.

2.3.2 Data Mismatch

Another common issue is data mismatch in data distributions between the training set and test sets or the training set and the real-world environment. Data mismatch tends to hurt the generalisation of trained predictive models.

Most commonly, data mismatch is caused by dataset shift (sometimes referred to as concept shift or concept drift) which happens when training and testing set distributions differ due to certain factors. Based on these factors, we can differentiate a few types of shifts: (i) population shift; (ii) annotation shift; (iii) prevalence shift; (iv) manifestation shift, and (v) acquisition shift [23].

Population shift means that the distribution of data attributes in the training set and test set differ. For example, let us suppose we are training a model to recognise the age of a person and we provide it with a training set where for most samples the attribute of age takes values between 10 and 40. The real-world environment distribution will of course be different with the age attribute having a much bigger range, meaning there is a population shift in our data.

Annotation shift can happen when there are multiple annotators working on a single dataset. This can cause differences in how the annotators label the same datapoint. These can result from individual annotator's bias or different policies or grading systems employed. The result would be annotation inconsistencies between multiple data generating centers.

Under **prevalence shift**, the differences between datasets relate to class balance between the training and test sets. This can arise for example from different predispositions in the training and test populations, or from variations in environmental factors.

Manifestation shift means that the way in which intrinsic attributes manifest – i.e. the attributes that indicate a specific target label – changes between domains. For example, the attributes which imply the presence of a disease are different in the dataset used for training than they are in real-world environment data. Manifestation shift makes it difficult for models trained on one domain to generalize to another, and it is generally not clear how to target the problem of manifestation shift.

Acquisition shift relates to data quality, in this case specifically the quality of images. Differences may be caused by the different instruments used to collect the images and their properties. For example, a scanning instrument at one hospital may be of older generation and produce images of low resolution while a scanning instrument at a different hospital may produce high-quality images. This could lead to acquisition shift.

2.4 Improving Performance on Biased Datasets

There exist many well-established methods which aim to counter bias in a dataset [6, 22, 24, 25]. We can divide them into data-driven approaches and model-driven approaches. For the first type, we try to manipulate the data prior to the training process so as to mitigate the existing bias. And for the second approach, the training objective gets modified accordingly so that the training process accounts for the existing bias. In this study, we focus mostly on the data-driven approach, however, other methods use the model-driven approach to good effect as well [26, 27].

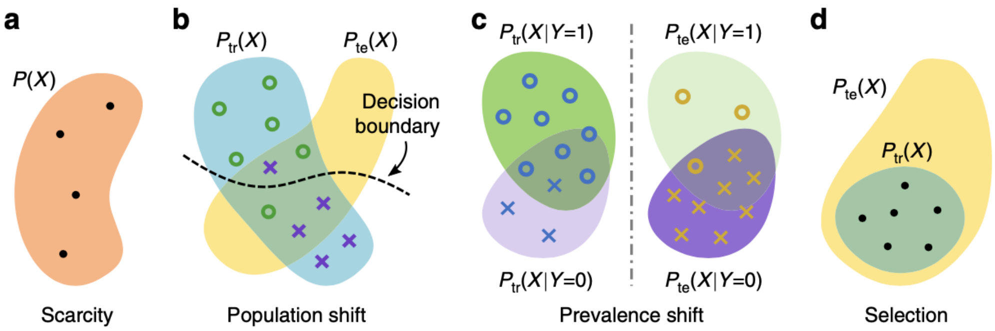

Figure 2.2: Different types of dataset bias. Taken from [23]

Some of the most-commonly used methods include resampling, re-weighing the loss function and data augmentations as well as many regularisation techniques. We go into more detail on each method used later in Chapter 3.

2.4.1 Data-driven Methods

Resampling is a technique used to balance the number of samples across classes in an imbalanced dataset in order to prevent the model from being biased towards the majority class. There are two types of resampling: (i) undersampling and (ii) oversampling. In oversampling, we duplicate the samples of the minority classes in order to increase their prevalence. In undersampling, samples of the majority classes are eliminated at random to reduce their prevalence. It is important to note that undersampling can often lead to information loss if not done carefully.

Data augmentation is a method which creates new data points from existing data by randomly applying controlled perturbations. Some common augmentations include rotation, flipping, cropping and adding noise. An important thing to mention is that data augmentation, when done in a controlled fashion, can be used to insert domain knowledge into the training dataset to improve the model's robustness to variation in held-out test sets. This is especially the case when we expect to observe certain perturbation at test time and would therefore like our model to be *invariant* to them.

Some data augmentation and regularisation techniques assume previous knowledge about the causes of poor model performance which can help tailor these methods to specific model needs. Next, we go through a few of these methods which have been proven to work well.

Cutout [22] is a data augmentation method which involves randomly masking out square regions of an image, typically of fixed size. Obscuring part of the image forces the model to learn the intrinsic features of the objects in the image, rather than relying on the background or peripheral attributes. Cutout is particularly useful when training object detection and segmentation models as it simulates occlusion and makes the model more robust to changes in the background. Additionally, it also helps to prevent overfitting, by introducing randomness in the training data.

Mixup is a data augmentation technique introduced in [24]. It is a simple method that generates new training samples by linearly interpolating pairs of input data and corresponding labels. The idea is to create new samples that lie on the convex combination of two original samples, by taking a weighted average of the input data and the labels. This allows the model to learn from the interpolated data, which helps to improve its generalisation performance and robustness.

Disentangled feature augmentation is a technique which uses latent vectors of image datapoints [5]. New samples are generated by swapping latent feature vectors of existing samples. An encoder and a linear classifier are trained to learn the disentangled representation of intrinsic attributes and bias attributes, respectively. These learnt vectors can then be swapped after several iterations of training have been performed to create novel datapoints.

2.4.2 Model-driven Methods

Re-weighing the loss function is a technique used to balance the contribution of different classes in a multi-class classification problem. In re-weighing, each sample is multiplied with a ratio equal to its population proportion over its sampling proportion to assign higher weights to the minority classes during training. The weight of each class can be set manually or it can be learned from the data.

Distributionally robust optimisation (DRO) is a method of regularisation aimed at minimising the worst-case training loss for predefined subgroups. Often, to generalise well, overparameterised models are used which help achieve high overall accuracy but fail to perform well on out-of-domain groups in test sets. For group DRO [26], spurious correlations in training data are defined which allows for constructing and dividing the data into subgroups. The aim is to minimise the loss of the worst-performing subgroup at each step during training. This approach achieves substantial improvements in worst-subgroup test set accuracy over standard approaches with minimal reductions in overall (average) test set performance.

2.4.3 Motivation: Debiasing Methods Discussion

While these methods of fighting dataset imbalance work to an extent, it is often unclear why exactly they improve the model's performance and why one method might perform better than others [6].

There have been studies trying to analyse the differences in efficiency of particular methods and find the underlying causes. For example, it has been found that resampling very often outperforms re-weighting. A 2021 study [28] shows that the difference in performance appears because of the stochastic gradient-type algorithms most often used for training models. Optimal learning rate selection for re-weighing is often challenging due to the noise of stochastic gradient algorithms. As part of the study, classification, regression, and reinforcement learning experiments are performed to empirically show that the findings are plausible.

Nevertheless, many of the commonly used methods simply lack good explanations and can even be counter-intuitive at times. Methods such as dropout or cutout attempt to boost performance by obscuring parts of the input images. Through this technique the method should "force" the network to focus on utilizing the full context of an input image rather than relying on specific visual features which might be peripheral attributes. This targets the problem of crucial objects or elements being occluded in test images which would require the network to perform recognition based on other, visible traits. In the cutout study [22], the authors initially intended to remove visual features with high activations in latter layers of a CNN, however when performing experiments they discovered that randomly obscuring square sections in input images gives similar results, while being significantly computationally cheaper. Naturally, the question is why does this approach work? Purposefully applying noise or removing parts of images in the training set seems like corrupting good data and feels counter-intuitive. Additionally, the final implementation where cutout obscures random sections poses even more questions as to why that proves to be more effective than obscuring specific regions which are highly recognisable by the model.

In this context, counterfactuals as a data augmentation method seem especially useful. It is clear why they would boost the performance of a model and they do that without throwing away any information or corrupting the existing data. In turn, they aim to generate plausible new input images that try to mimic data which we would be likely to find in potential test sets or real-world testing scenarios. Furthermore, if randomly sampling counterfactuals improves the classifier's performance then potentially only very little data analysis needs to be done prior to successful debiasing.

Of course, counterfactual data augmentation makes use of generative models which require significant resources and can be time consuming to train. This is something we are not forced to take into consideration when using more standard debiasing methods.

2.5 Background: Counterfactuals

2.5.1 Causality and Structural Causal Models

In causality, Pearl's ladder of causation is a concept used to represent a three-level hierarchy of problems of increasing difficulty. The three levels are **association**, **intervention** and **counterfactuals** [29].

The lowest level, **association**, entails the existing correlations in data in the system. Meaning that we want to answer questions like "given that some variables are observed to be x what is the probability that some other variable is equal to y".

The middle level, **intervention**, is about causal relationships. We perform some intervention on the system and want to predict what the effects of our actions are. For example, if we force a variable to be equal to x what is the probability that some other variable is equal to y.

The highest level, **counterfactuals**, is about the causal relationships between variables and the effects of modifying some variables on others retrospectively. Which means we ask questions like, given that variables $Z \subseteq V$ were observed to be z, if variables $X \subseteq V$ were forced to be x then how likely is it that variables $Y \subseteq V$ would have been equal to y?

To model causal mechanisms and relationships of a system we can use **Structural Causal Models (SCMs)**.

Structural Causal Models comprise a structure similar to directed acyclic graphs (DAGs) and consist of three elements:

1. A set of **variables** describing a causal system. There are two types of variables: (i) observed or endogenous, which are measured in our data; and (ii) unobserved or exogenous, considered background conditions for which we have no explanatory mechanisms.

2. **Relationships** between variables which determine how values are assigned to each variable in the system. For example, $x_i = f_i(\mathbf{pa}_i, \epsilon_i)$ describes a process by which variable x_i is assigned its value through a structural assignment or causal mechanism f_i. This mechanism is a function of the variable's parents $\mathbf{pa}_i$, which is the set of its direct causes, and an exogenous factor ϵ_i.

3. **A probability distribution** over unobserved (exogenous) variables describing the likelihood of these variables taking on a specific value.

In SCMs represented by DAGs, all the relationships are one-directional from cause to effect, and no variables can have a causal relationship with themselves as that would induce a cycle or feedback loop.

A counterfactual query, e.g. the probability of a counterfactual outcome $P(Y_x = y \mid \epsilon)$ is the probability that Y is y if X were x given the exogenous noise ϵ. Counterfactual inference of this query entails a three-step process of abduction, action and prediction (more detailed description in 2.5.3).

Relationships in an SCM can be represented by a directed acyclic graph with edges pointing from causes to effects, which we call a causal graph. Using causal graphs, it is then possible to perform actions or interventions and compute causal effects. We can use deep generative models such as GANs, VAEs or diffusion models to learn the mechanisms in an SCM from observational data.

To consider an example lets look at a case similar to “Case Study 1: Morpho-MNIST” from [18]. Here we use the Morpho-MNIST dataset which is a modification of the MNIST dataset of handwritten digits [21] with additional “thickness” and “intensity” attribtues introduced using the Morpho-MNIST framework [30]. A causal relationship is introduced where the thickness of a digit influences the brightness (intensity) of the image. This could be written as follows:

$$t = f(\epsilon_T)$$

$$i = f(\epsilon_I; t)$$

$$x = f(\epsilon_X; i, t)$$

where ϵ_T and ϵ_I are sampled from some probability distribution and ϵ_X is sampled from the set of digits as defined in the MNIST dataset.

A setup like this can then be used to train a model which can generate images following these causal relationships. In [18] such a model is trained and compared with two other models which do not exploit this causal structure. Some examples of the generated images and their originals can be seen in Figure 2.3.

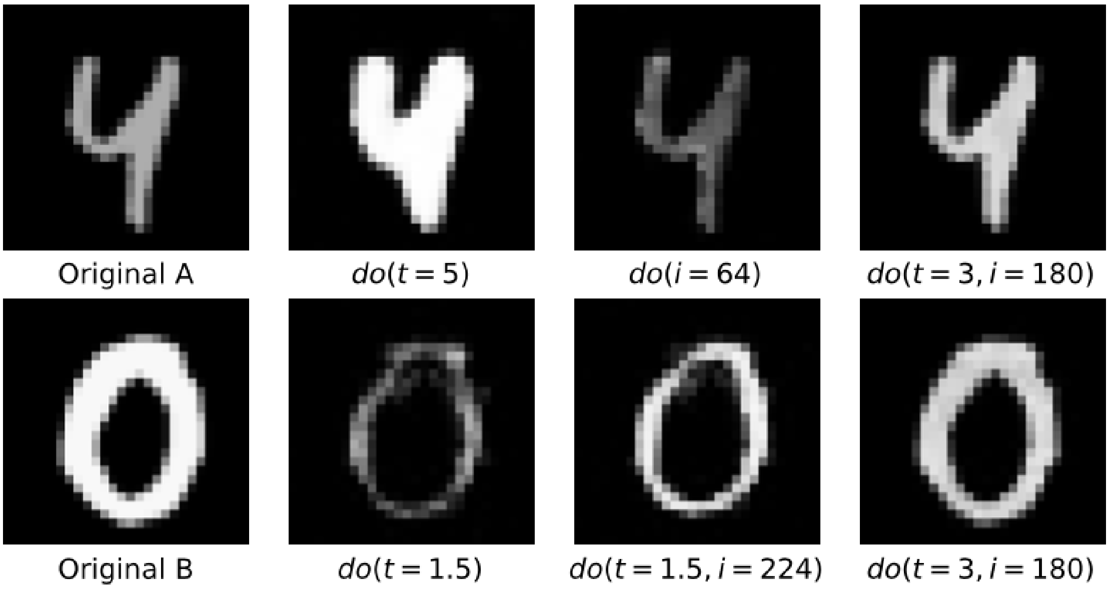

Figure 2.3: Original and counterfactual images generated by the full model. Here $do(...)$ represents an intervention on t - the thickness or i - the intensity. Taken from [18].

2.5.2 Generative Models

Variational Autoencoders

A variational autoencoder (VAE) is a probabilistic variant of a deterministic autoencoder which comprises an encoder and decoder model. The encoder, implemented using neural network layers, is trained to learn a meaningful representation of input data as a distribution over the latent space (encoded space). The encoder typically performs dimensionality reduction on the input data, extracting relevant features, to encode it in the latent space. Using the decoder, implemented using neural network layers as well, we can generate new data by forwarding samples from the encoded latent space. The objective used during training takes into account two factors. The first one is the minimisation of the reconstruction error, which measures the difference between original and reconstructed data. The second terms ensures the regularisation of the learnt latent space using Kulback-Leibler divergence between the learnt distribution and a standard Gaussian prior distribution. This ensures that the latent space is constrained and enables sampling new datapoints via the prior.

Normalising Flows

Normalising flow models are based on a series of bijective functions. They aim to model complex probability distributions of data by successively applying invertible functions to a simple base distribution such as a standard Gaussian. They are trained using negative log-likelihood loss.

2.5.3 DSCM Architecture

A Deep Structural Causal Model (DSCM) is an SCM that uses deep-learning components to model the causal mechanisms [18]. These deep-learning techniques are introduced into SCMs to work for more complex, higher-dimensional data such as images. [18] propose using normalising flows and variational inference to perform the three necessary steps mentioned above for counterfactual inference.

Therefore, the steps for deep counterfactual inference are as follows:

1. **Abduction:** the goal of abduction is to predict the exogenous noise ϵ given observed evidence. The abduction can be performed independently for each conditional mechanism given its corresponding observed variable and its parents. Depending on whether a given mechanism is invertible or not we can obtain the noise using different methods. For invertible mechanisms, the noise can be obtained by simply inverting the mechanism f such that $\epsilon_i = f_i^{-1}(x_i; \mathbf{pa}_i)$, where x_i is the observed variable and $\mathbf{pa}_i$ are its observed parents. When the mechanism is not invertible, a little care needs to be taken. For example, for mechanisms based on implicit amortised likelihood where a trained encoder is used, the noise can be approximated by $\epsilon_i \approx e_i(x_i; \mathbf{pa}_i)$, where e_i is an encoder model.

2. **Action:** in the action step the causal graph is modified to reflect the desired intervention and each variable x_i we wish to intervene on is replaced either by a constant $\widetilde{x}_i$ or by a new mechanism $\tilde{f}_k(\epsilon_k; \widetilde{\mathbf{pa}}_k)$.

3. **Prediction:** in this step we can sample from the new, modified structural causal model to determine the counterfactual of the observed variables of interest.

2.6 Evaluation

2.6.1 Evaluating Counterfactuals

Evaluating counterfactuals themselves is a very hard task. In situations where we are interested in generating counterfactuals we normally do it because we lack certain data therefore there is no target output to compare to, which poses a big challenge. There is currently ongoing research on how to evaluate the generated data points. Performance analysis can be done, for example, based on three axioms, namely effectiveness, composition and reversibility [31] defined as:

- **Effectiveness:** performing an intervention on a variable x to have a specific value will actually cause the variable to have that value.

- **Composition:** performing a null transformation (i.e. an intervention which does not change the value of a variable) will not have any effect on other variables in the system.

- **Reversibility:** given an invertible mechanism and an observation x, its direct causes (parents) $\mathbf{pa}$ and its counterfactual x^*, we have that, if $x^* := f(x, \mathbf{pa}, \mathbf{pa}^*)$ then $x := f(x^*, \mathbf{pa}^*, \mathbf{pa})$, where f is a counterfactual function which is a mapping between an observation and a counterfactual.

Using those axioms we can determine how good a given approximation of a counterfactual function is.

For this project however, it will not be necessary to show that the counterfactuals themselves are plausible and authentic. This is because we want to use them primarily to train classification networks downstream. Therefore, the evaluation should focus on how well we can train the networks and how efficient the method is in debiasing the training set. With that said, it is likely that more realistic counterfactuals will perform better at debiasing.

2.6.2 Evaluating Predictive Models

There exist many evaluation metrics commonly used for assessing the performance of a machine learning model on a specific task. Most-commonly during testing a confusion matrix is devised with its elements representing the correctly and incorrectly classified datapoints. Multiple metrics can be derived from the confusion matrix such as accuracy, precision, recall etc. It is however very important to choose an adequate metric when evaluating a machine learning model. Some metrics can be very misleading when used incorrectly and can lead to constructing sub-optimal models.

For models trained on balanced datasets, using accuracy as an evaluation metric is a good choice and is common practice. Problems with using accuracy arise when the training dataset is biased as for example in a classification problem with imbalanced class distribution. Accuracy assigns a bigger weight on the majority classes and a much smaller weight on the minority classes [32]. Therefore, to correctly assess the performance of a model, other metrics which take into account class imbalance or bias ought to be used.

One such metric is the **F1-score**, (equation (2.1)) which is a harmonic mean of precision and recall which allows it to account for the size of each class. F1-score penalizes models that achieve high accuracy by predicting the majority class all the time:

$$F_1 = 2 \cdot \frac{precision \cdot recall}{precision + recall}. \tag{2.1}$$

Another popular metric which can be used for imbalanced datasets is the **AUC-ROC curve**. ROC is a measure of probability and AUC measures separability. Therefore, the metric measures how much the model is capable of distinguishing between different classes [33]. The basic ROC can be used for binary classification and to use the metric for multi-class problems we calculate it by taking the average of AUCs calculated for all classes against each other.

To calculate the AUC-ROC curve we use **True Positive Rate** (equation 2.2), also referred to as sensitivity, and **False Positive Rate** (equation 2.3), also referred to as fall-out or false alarm ratio. However, these metrics can also be used on their own as they provide valuable information and are insensitive to dataset bias.

$$TPR = \frac{TruePositives}{Positives}, \tag{2.2}$$

$$FPR = \frac{FalsePositives}{Negatives}. \tag{2.3}$$

Another metric commonly used is the **Kappa coefficient** (equation 2.4) which has its origins in psychology [34]. It measures the agreement between two evaluators who rate certain subjects. It is used in machine learning to give a score of how a model is performing in comparison with an untrained model (so one which chooses at random), taking into account the imbalance in classes.

$$\kappa = \frac{p_0 - p_e}{1 - p_e} \tag{2.4}$$

where p_0 is the overall accuracy of the model and p_e is the measure of agreement between the model's predictions and the predictions made by the untrained model.

With that said, every situation is different and it is crucial to tailor the evaluation metrics to a concrete setup. As previously mentioned, data bias can have many causes and can take many forms. For a simple class imbalance looking at the F1-score might be enough. However, when the bias is based on attribute correlation, or if we are dealing with acquisition shift for example, these metrics might not be enough to show the true performance of the model on real-world data.

2.7 Related Work

There is a lot of research around the evaluation of counterfactuals as a bias mitigation method in the NLP area [35, 36]. In the domain of computer vision, this concept is still relatively new and hence there is less research done on it. There are however some case studies around evaluating bias mitigation using counterfactual images.

2.7.1 Counterfactual bias mitigation

In [37] a bias mitigation method is described, which uses counterfactuals to modify the model training objective. Additionally, bias evaluation is performed on the CelebA dataset [38] using counterfactual images as well as fairness analysis, utilising counterfactuals generated around specific, bias attributes to compare the predictions of a trained classifier and estimate its bias. In the experiments, an alternative method is used, called *ImageCFGen*. The proposed architecture consists of an encoder, a generator (decoder) and Attribute-SCM which is a component which performs interventions on the desired attributes.

The authors of [39] introduce another alternative conditional generative model called CounterSynth, which they use to generate counterfactual augmentations of training data. They run experiments, using the UK Biobank dataset, where they compare standard ERM methods with group DRO and counterfactual augmentation. In their work they focus on identified bias attributes and compare both global and local performance. They then analyse the performance of multiple predictive models trained to label test images using those chosen attributes.

Similarly, [40] analyse attribute confounding and propose an algorithm for counterfactual data generation where they aim to remove the biases based on their measurements. They conduct a comparison of standard empirical risk minimization methods and the novel causality-based augmentations.

2.7.2 Counterfactual fairness estimation

Counterfactual augmentation is also used to measure classifier fairness in [41]. They measure the fairness of computer vision APIs widely used both commercially and in research, defined using the following equality: $P(Y_{A \leftarrow a}(x) = y \mid x) = P(Y_{A \leftarrow a'}(x) = y \mid x)$ where $A \leftarrow a'$ indicates an intervention. If the equality holds for all y, a and a' then a classifier is deemed fair.

To perform the analysis they use generative models to obtain image counterfactuals around certain common bias attributes such as "race". They find that some of those APIs behave in a way which is related to the gender gap/distribution present in many occupations.

Chapter 3

Bias Mitigation

In this chapter, we describe all the standard bias mitigation methods we use as baselines for performance improvement comparison (Section 3.1.1).

We introduce the counterfactuals-based methods including: (i) dataset expansion with targeted interventions; (ii) dataset expansion with random sampling; (iii) counterfactual regularisation, and (iv) several hybrid methods combining counterfactual data augmentation and standard methods (Section 3.1.2).

Lastly, we provide implementation details on all methods used (Sections 3.2, 3.3).

3.1 Debiasing Methods

3.1.1 Baselines

To enable comparisons between methods, we debias using five different standard techniques:

- **Sampling.** In the experiments we make use of oversampling as the data available is scarce to begin with, therefore it does not make much sense to undersample. By counting the number of samples per each class or predefined subgroup we calculate their difference in size and then oversample by that difference. Multiple attribute subgroups can be oversampled at once until the underrepresented subgroup is as numerous as the others. For Morpho-MNIST we oversample based on the class label. For Mimic CXR we oversample both around the class label and around the specific targeted attributes chosen for a given experiment (more details in Chapter 6).

- **Loss Function Re-weighing.** This is done where applicable. It is applicable when we are dealing with an undersampled class like in the example of Mimic CXR where there is significantly less data of the negative class.

- **Standard Data Augmentation.** For the Morpho-MNIST dataset we use blurring and adding noise (Gaussian and Salt&Pepper nosie) and rotations by a

small angle (less than 30 degrees).
For the Mimic CXR dataset we use all the augmentations used for Morpho-MNIST plus flipping, as flipping makes more sense in the context of medical images but not much sense for digit images.
An augmentation is chosen at random from a list of available options for each datapoint. We apply the augmentations to the targeted undersampled class or subgroup until it is as numerous as the others or as the most numerous subgroup.

- **Group DRO.** For the Mimic CXR dataset we define the DRO groups around a specific attribute and calculate the counts of each group appropriately. To calculate the DRO loss we adapt the original algorithm as defined in [26]. First, per-sample losses are calculated using the criterion loss function (Cross Entropy), these are then sorted into the defined groups and average loss is calculated for each group. The total, returned loss calculation is based on the per group losses and follows Algorithm 1.

- **Mixup.** We also use the previously mentioned mixup method [24]. Pairs of datapoints are combined online, during training. Pixels of both datapoints are combined using a λ hyperparameter and both labels are used for loss calculation which follows Equation 3.1.

$$l = \lambda \cdot \mathrm{loss\_fn}(\mathrm{logits}, \mathrm{targets_a}) + (1 - \lambda) \cdot \mathrm{loss\_fn}(\mathrm{logits}, \mathrm{targets_b}) \tag{3.1}$$

where logits is the output of the model for the combined pixels input. Mixup is a method which combines datapoints randomly hence no specific attribute bias is targeted in our study.

Algorithm 1 Group DRO Loss

```
  Require: $y_{pred}$, $y_{true}$, group_idxs
  Initialize: $\mathrm{loss_0} \leftarrow \left[\frac{1}{\mathrm{n\_groups}}\right]_{\mathrm{n\_groups}}$
$\mathrm{loss\_per\_sample} \leftarrow \mathrm{loss}(y_{pred}, y_{true})$
$\mathrm{group\_losses} \leftarrow \mathrm{get\_loss\_average\_per\_group}(\mathrm{loss\_per\_sample}, \mathrm{group\_idxs})$
$\mathrm{loss_t} \leftarrow \mathrm{loss_{t-1}} * \exp(\eta * \mathrm{group\_losses})$
```

We then treat these methods as baselines for the new counterfactual data augmentation method.

3.1.2 Counterfactual Inference Methods

In the context of using counterfactual generation as a data augmentation method, we can distinguish two types of augmentations based on their influence on the target label of a data point:

- **Label-preserving** perturbations are ones which do not change the target label. For example, in the context of digit classification on the MNIST dataset, generating images by altering the thickness of a digit is a label-preserving perturbation as it does not modify the digit label.
- **Label-altering** perturbations modify the target label as a result of the augmentation they perform. Again, for example, in the context of digit classification on the MNIST dataset, generating images by changing the digit while keeping, for instance, the thickness as is, is a label-altering perturbation as the digit label changes.

Distinguishing between those two types of augmentations is important when planning to debias a dataset using counterfactuals and here we use both types.

We attempt to achieve performance improvement and generalisation via generative models in two ways: (i) data manipulation prior to training; (ii) by modelling the training objectives appropriately [37]; as well as several hybrid approaches combining counterfactual data augmentation and different standard methods:

- **Expanding the dataset prior to training**
  For the first approach, the original dataset is expanded with counterfactual images. The images are generated by performing interventions on specific variables and to determine which variable to intervene on dataset analysis is necessary. For each dataset we compare the impact of debiasing with counterfactuals with respect to several different attributes. For the Morpho-MNIST dataset the attributes we intervene on are “thickness” and “digit label”, and for the Mimic CXR dataset the attributes are “sex”, “age”, “race” and “disease label”. All counterfactual images are generated from training samples and we generate them until the underrepresented groups are as numerous as the largest group.
- **Expanding with random sampling**
  The second approach is a slight modification of the first one. Rather than modifying a specific attribute by tailoring the augmentation to a predefined bias (e.g. generating young female samples as the existing data shows a deficit thereof) we make use of random sampling and generate additional counterfactual data at random. This approach does not require much prior knowledge about the dataset hence in certain constrained scenarios it can be superior. Here it is unclear how many counterfactuals to generate therefore we aim to generate enough to bridge the gap between the class counts.
- **Expanding the dataset & mixup**
  For this method we use targeted counterfactual generation to expand the dataset and in addition use mixup to bridge some potential gaps in attribute distributions with combined examples.
- **Counterfactual regularisation**
  The next approach uses counterfactual regularisation [37] and is a model-driven approach. It enforces that the classifier predicts the same output for

both the original and counterfactual image. A regulariser term, calculated from the output of the classifier before applying the sigmoid activation function, is added to the loss to train the classifier:

$$loss = CrossEntropyLoss(logits, y_{true}) + \lambda * MSE(logits, logits_{cf}), \qquad (3.2)$$

where $logits$ is the output of the classifier for the original images and $logits_{cf}$ is the output of the classifier for the counterfactual images, before applying the sigmoid activation function, y_{true} are the ground truth labels and λ is a hyperparameter.

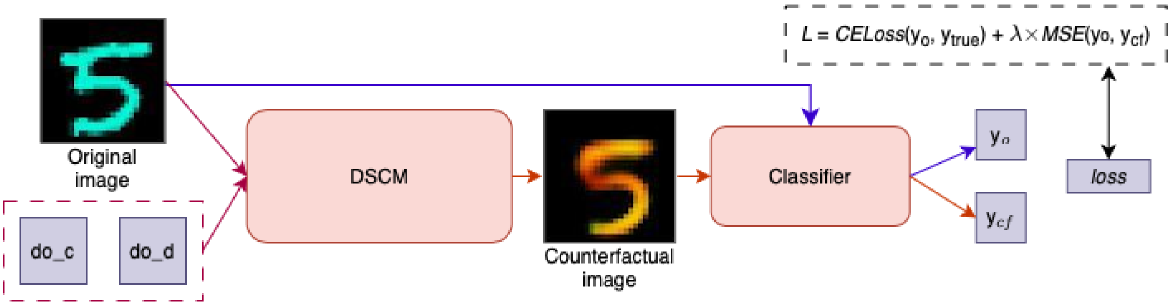

Figure 3.1: Counterfactual regularisation pipeline.

3.2 Baseline Methods: Implementation

The oversampling method was implemented from scratch as we needed to oversample not only on the level of classes but also individual attributes, taking into consideration attribute correlations.

For data augmentations *torchvision* [42] and the *Pillow* library [43] were used. For each sample an augmentation is chosen at random from a list of available augmentations. As mentioned, for group DRO we adapted the original algorithm as proposed in [26].

Similarly, for mixup the proposed algorithm was used for online interpolation of samples.

3.3 Counterfactual Generation

For the generation of image counterfactuals, we trained checkpoints of DSCMs for Morpho-MNIST, Colored-MNIST and Mimic CXR datasets. For dataset expansion, the counterfactuals were generated prior to training the model from observational data, to enable a faster training process. Consequently, the same counterfactuals are used in each epoch.

For CF regularisation, the counterfactuals are generated online for each batch of training data to use a fresh batch on every epoch. They are then passed through the model so that original and counterfactual predictions can be compared and used in the loss function.

Chapter 4

Datasets

In this chapter we explain the requirements we took into consideration when choosing the datasets for bias evaluation and mitigation comparison (Section 4.1).

We introduce the chosen datasets and the particular tasks which we focus on. We also go into details on the necessary data preprocessing steps implemented for each dataset (Sections 4.2, 4.3, 4.4).

4.1 Requirements

In order to evaluate the impact of different debiasing methods on the performance of image predictive models we require that the datasets used consist of annotated images with discrete labels as we focus primarily on classification tasks and utilise supervised learning. It is not important for this project whether the labels are binary or not.

It is preferable that the datasets already exhibit some kind of bias. This could mean that certain label classes are under or over represented, that specific attribute subgroups are correlated either between each other or with the target labels or any other type of bias. However, this is not strictly necessary, as in some cases it can be straightforward to synthetically introduce bias which is what we do for some of the datasets.

4.2 Morpho-MNIST

The Morpho-MNIST dataset is an extension of the MNIST dataset of handwritten digits. There are 10 classes representing the 10 digits. The dataset consists of 70k images, 7k per each class. We split it into a training set of 60k images and a test set of 10k images. In addition to a class label the Morpho-MNIST dataset also consists of two additional attributes describing each sample, namely “thickness” and “intensity”. The values of these attributes are continuous within specified ranges.

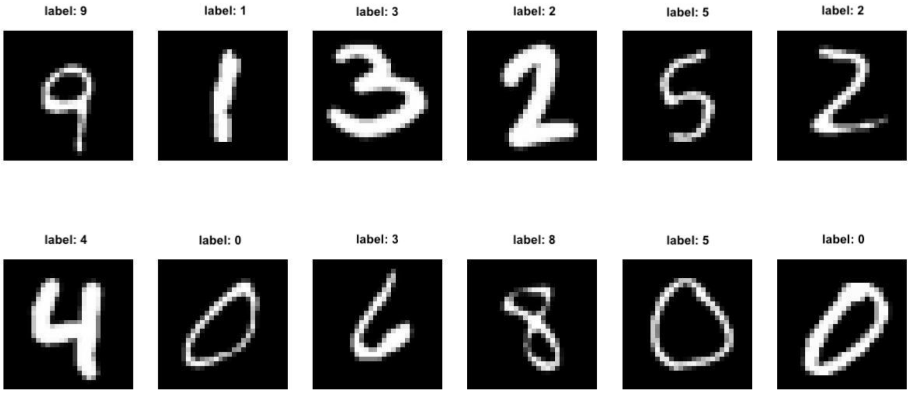

Figure 4.1: Example images from the Morpho-MNIST dataset.

As the Morpho-MNIST dataset itself is mostly equally distributed in terms of all the attributes, we introduced a bias ourselves to get baseline results which we could then try and improve upon with different methods. The bias which we introduce is based on the "thickness" attribute. To change the thickness of a digit we use the Morpho-MNIST framework [30] which allows for multiple perturbations specifically of the MNIST dataset. We define a classification of classes into those which are more likely to consist thicker digits (1, 3, 6, 9) and others which consist mostly of thin digits (0, 7, 8). The remaining digits are equally likely to be thick or thin. We then construct the biased dataset with varying percentages of bias-aligned samples. The average thickness for each class is depicted in Figure 4.2.

To synthetically inject bias we introduce the concept of the percentage of bias-conflicting samples. Given a specific bias, we modify the dataset so that the bias-aligned and bias-conflicting samples amount corresponds to the percentage. For example, in a colored version of the MNIST dataset where each digit is tied with a chosen color, e.g all ones are red, all twos are blue, we set the percentage of bias-conflicting samples to 5%. Then 95% of all images will abide by the bias, meaning 95% of all ones will be red, 95% of all twos will be blue etc. The remaining 5% of all images is colored with a randomly chosen color.

We experimented with different values of the bias-conflicting samples percentage. We tried 1%, 2% and 5%. With 2% and 5% the results were not satisfactory as the classifier was able to easily train from the scarce data available and still obtain good results. Therefore, we tried with 1% of bias-conflicting samples and observed a drop in overall accuracy of around 10%.

For the test set all of the classes are balanced and there is the same amount of thick and thin samples per each class.

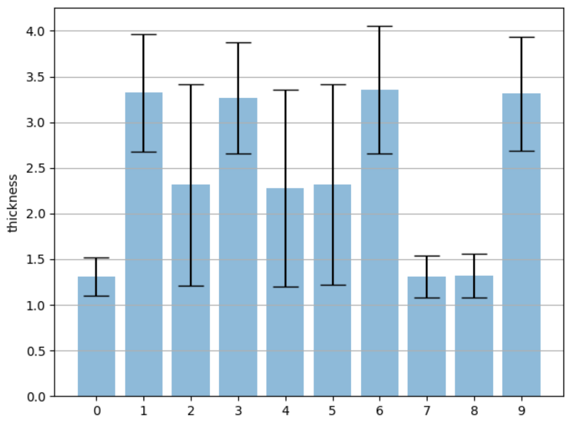

Figure 4.2: Thickness of biased Morpho-MNIST

4.3 Colored-MNIST

Colored-MNIST is another extension of the MNIST dataset, where we assign a distinct color to each digit and apply it on the foreground of each image (Figure 4.3). The dataset consists of 60k training images and 10k test images. The test set is fully balanced, meaning each digit receives a random color out of the ten available. Again, we make use of the bias-conflicting percentage variable and focus on two scenarios. One, where 1% of the images are bias-conflicting (colored randomly) and another where all samples follow the bias. The later scenario is particularly interesting from the perspective of utilising counterfactual data augmentation, as it would be very challenging to mitigate the bias using any of the baseline methods. It is, however, a representation of many real-world scenarios where certain data is completely unavailable during train time.

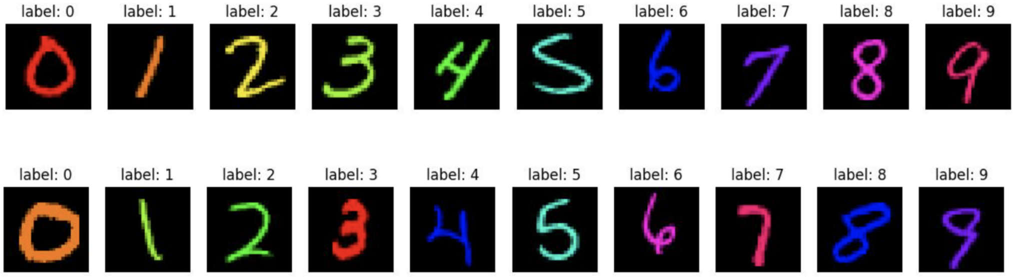

Figure 4.3: Example images from train (top) and test (bottom) Colored-MNIST data.

4.4 Mimic CXR

Mimic CXR [44] is a dataset of approximately 200 thousand images of chest x-rays and attributes corresponding to each image. For the purposes of this project, we use the “sex”, “race” and “age” attributes as well as all attributes associated with a disease to determine the value of a binary class – “finding” or “no finding”. Here we focus on a single disease – Pleural Effusion. Therefore“finding” means the subject was diagnosed with Pleural Effusion and “no finding” means they were diagnosed as being healthy.

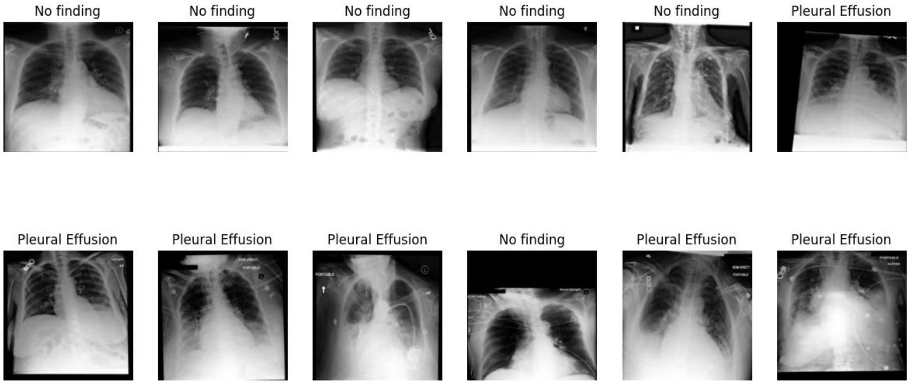

Figure 4.4: Example images from the Mimix CXR dataset.

As mentioned, we filter out many samples from the original dataset to suit the tasks. After this procedure there are approximately 100k images left. We divide the data into training, validation and test sets with a 60:10:30 split ratio, which gives around 60k images in the training set, around 10k in the validation set and 30k in the test set. We resize all the images to a 192x192 resolution. The “race” and “sex” attributes are discrete values and the “age” attribute is continuous therefore we define five age ranges, every 20 years, to target age bias.

The dataset contains a number of attributes describing each image. From analysing the dataset, we can see there is a strong bias correlated with the attributes distribution. It is clear that there are significantly more negative samples, meaning those of healthy patients (Table 4.1). While this difference is not very significant for some subgroups - like race “White” - it makes a big different for other, e.g. the ages 0-19, 20-39 or race “Black”. Unfortunately, this bias is also present in the validation and test set which makes fair testing challenging. One way around this would be to downsample majority classes in the validation and test sets however there are so few samples in the minority classes that this is rather impractical. Alternatively, we can look at metrics which do not take into consideration subgroup sizes or look at relative changes in subgroup performance between two models.

There is also some attribute correlation bias in addition to data scarcity. The younger patients (aged 0-39) are much more likely to be healthy than diagnosed with Pleural Effusion, while among the older patients (aged 60-100) there are more samples with the disease present. The middle-aged group seems to be more balanced. This irreg-

	Positive samples	**Negative samples**
	Sex	
male	**55%**	**53%**
female	45%	47%
	Race	
White	**85%**	**75%**
Asian	4%	4%
Black	11%	21%
	Age	
0-19	0.07%	0.3%
20-39	4%	9.8%
40-59	20.17%	28.9%
60-79	**47.91%**	**42.6%**
80-100	27.85%	18.3%

Table 4.1: Attribute percentage counts per class in training set

ularity may condition a classifier to associate the target label with the age attribute.

Further, the size imbalance continues within certain attributes. For the “race” attribute, there are significantly more White patients and Asian patients are the least numerous group. The “sex” attribute is by far the most balanced one and as expected within the “age” attribute there are much more samples of middle-aged patients (40-80) and there is very little data of young individuals (0-40).

Chapter 5

Experimental Setup

In this chapter, we introduce the setups and tasks chosen as well as the baselines used for further comparison of the previously described bias mitigation methods (Section 5.1).

We also describe several additional evaluation techniques which were used in the experiments for performance comparison (Section 5.2).

5.1 Models & training

5.1.1 Morpho-MNIST

For Morpho-MNIST we define a multiclass classification problem of digit prediction.

First, we train a classifier on the biased dataset with the "thickness" attribute correlated with the digit and certain classes undersampled (labels 0, 1, 3, 6, 7, 8 and 9 are undersampled where their counts are cut down to 20k). We experimented with different values of the bias-conflicting percentage variable and finally settled on using 1%.

The thickness bias is visible on the mean thickness diagram (Figure 4.2). We calculate the overall performance and per-class performance as well as performance based on the value of the "thickness" attribute. The overall accuracy reached is 89.7% with F1-score of 89.5%, while the SOTA performance on balanced Morpho-MNIST is around 99% accuracy. From per-class performance (Table 5.1) we can see certain classes dropped in performance more than others, mainly we notice worse performance for the undersampled classes with thickness bias. Classes left intact, such as label 4 and label 5, show the best performance except for label 0 and label 1 which are arguably the easiest to distinguish.

	Precision	Recall	F1-score
label 0	91.23	94.43	92.12
label 1	99.14	96.3	97.74
label 2	84.22	99.54	91.04
label 3	88.71	90.15	89.34
label 4	90.07	99.31	94.63
label 5	88.65	98.77	93.45
label 6	96.31	80.12	87.65
label 7	86.74	77.79	81.27
label 8	91.24	82.34	87.66
label 9	86.66	80.11	83.75

Table 5.1: Baseline Morpho-MNIST digit classification per-class performance.

5.1.2 Colored-MNIST

For Colored-MNIST, similarly to Morpho-MNIST, we define a multiclass classification task of digit prediction. From training a baseline classifier it is obvious the model's predictions align with the digit-color correlation present in the dataset. The performance is fairly equal among classes. For the fully-biased version of the dataset (0% bias-conflicting samples) there is a significant reduction in performance (from accuracy of 73.53% to 19.29%) which is expected as we are fully correlating the label with the color attribute.

5.1.3 Mimic CXR

We define two tasks on the Mimic CXR dataset: disease classification and race classification.

Disease Classification

From investigating the original dataset, it is clear there is a strong bias in the data, especially in terms of race and age of patients. We calculate the performance per class and also per attribute for each task attempted. Table 5.2 shows some standard evaluation metrics for the disease classification task. Both accuracy and F1-score are similar which would indicate the model performs comparably for both labels and is unbiased. That is why it is important to look at per-attribute performance as well and analyse whether the classifier is not biased with respect to certain subgroups. We observe the model achieves higher results for samples marked as "White" than it does for "Black", however, it is performing extremely well for samples marked "Asian", which could be because there are only a few Asian samples in the test set. Similarly there is a significant drop in performance for very young and elderly patients. The sex attribute is the most balanced one and as expected we do not observe a big difference in performance between male and female patients.

	Accuracy	F1-score	Precision	Recall
	Disease classification			
No finding	90.12	90.33	91.65	89.05
Pleural Effusion	89.04	88.68	87.22	89.63
macro-avg	89.58	89.51	89.64	89.57
	Race classification			
White	92.36	92.12	91.88	92.36
Asian	42.32	43.37	44.47	42.34
Black	72.62	73.04	73.46	72.62
macro-avg	86.65	69.51	69.94	69.1

Table 5.2: Baseline disease/race classification global performance.

	Female	Male	0-19	20-39	40-59	60-79	80-100
Accuracy	84.31	88.84	87.65	82.55	87.3	87.72	85.35
F1-score	89.08	93.31	90.74	86.77	91.23	92.31	91.4

Table 5.3: Baseline race classification per-attribute performance for sex and age attributes.

Race Classification

For the race classification task, we analyse the data again and performance of a baseline classifier. As previously mentioned, the data is highly imbalanced with respect to the race attribute. The trained model achieves very good performance for the largest subgroups of White patients and the lowest performance for Black patients as this subgroup is significantly under represented in the training set (Table 5.2). Further, classification of patients from different age ranges shows different results, with middle-aged patients achieving higher accuracy than young and older patients. Further, male patients score slightly higher results than female patients (Table 5.3).

5.1.4 Training Details

Generative Model

The DSCMs used for this project consist of two components trained separately. For all variables other than the image, normalising flows were trained. A VAE is used for the causal mechanisms of the images.

For each dataset we trained a DSCM to generate plausible counterfactuals. Each DSCM is trained on a biased version of the dataset to simulate the accessibility of

data in real-world scenarios where data available to train GANs would be imbalanced by definition of the problem.
For the Morpho-MNIST dataset, the attributes learned by the DSCM are “thickness”, “intensity” and “digit”. For the Mimic CXR dataset, the discrete attributes are “sex”, “race” and “finding” and there is a continuous attribute “age”. The normalizing flow-based model and the VAE are both trained separately. They are trained for 200 epochs each. Once all the individual components are trained, they are combined into one module which can then be used for inferring visually plausible counterfactuals.

The DSCM is trained solely on the training sets to simulate a situation where test data is truly unseen and unavailable. The training sets are therefore split into train and test sets using an 80:20 ratio.

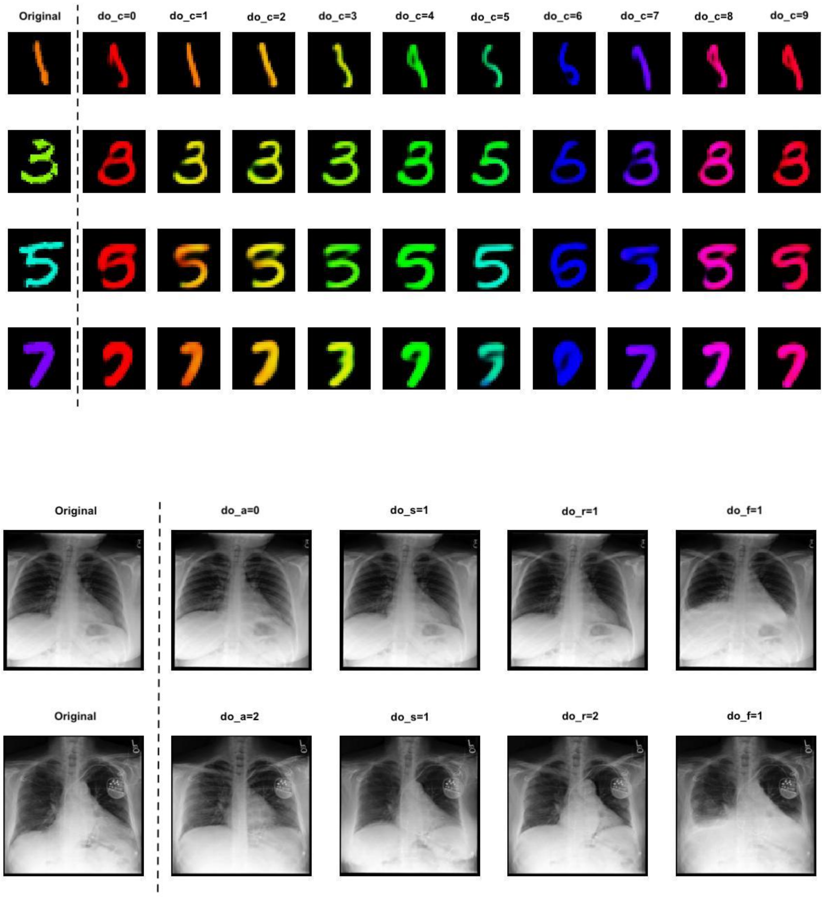

Figure 5.1: Top: example original images sampled from the Colored-MNIST training set and counterfactual images generated using a trained DSCM by applying interventions on the color attribute. Bottom: example images sampled from Mimic CXR and counterfactual images generated by applying interventions on age, sex, race and disease, respectively.

Predictive Models

For Morpho-MNIST and Colored-MNIST we fine-tune a pre-trained ResNet18 model with an extra linear layer. We use the AdamW optimiser and start learning rate of 0.0003 with weight decay of 0.05 and train for 20 epochs with early stopping and a batch size of 32.

For Mimic CXR, first, we train a classifier without applying any modifications, to get a good reference for all further experiments. We first used a ResNet architecture [8] with an additional linear layer, as it can be trained with a large number of layers easily without increasing the training error at the same time.

Then we trained using a DenseNet-121 architecture to compare the results, and we achieve better performance with DenseNet. Therefore, we will be using the DenseNet architecture in the experiments. Furthermore, [19] published their results on the Mimic CXR dataset using DenseNet-121 so it was possible for us to compare our results with theirs and use them as a benchmark. They report an AUC-ROC score of 0.83% on a multiclass classification problem. As our task is a binary classification we expect to achieve slightly better results overall.

We use the AdamW optimiser and start with a learning rate of 0.00005 for disease classification with weight decay of 0.05. We train for 20 epochs with early stopping.

We use a batch size of 32 as that was the maximum possible batch size on the machines available to us to avoid out-of-memory errors.

5.2 Evaluation Methods

5.2.1 Standard Metrics

Multiple metrics were used to evaluate the experiments done as part of this project. Standard metrics, mentioned before in Chapter 2, such as accuracy, precision, recall or F1-score are used to compare both overall and per-class performance. However, a fair model should ideally approach the maximum achievable performance equally closely across all subgroups [45]. This definition means that standard metrics, most often used in evaluating machine learning tasks, are not up to standard here. It is therefore necessary to also analyse the model's performance comparing the different subgroups rather than just looking at the model as a whole.

5.2.2 Global & Local Performance

It is trivial to achieve equal performance over subgroups by lowering the overall performance to that of the worst-performing subgroup, however, we do not want to harm global performance in the process. Therefore, the goal is to increase the performance over some predefined subgroups while maintaining a good baseline level.

To focus on the overall change, we can calculate the global (ΔG) and local (ΔL) changes in performance defined in 5.1 and 5.2 respectively:

$$\Delta G = \frac{\sum_k (a_k^{new} - a_k^{base})}{\sum_k a_k^{base}} \tag{5.1}$$

where k is the number of subgroups, a_k^{new} is the mean performance of subgroup k for a chosen debiased model and a_k^{base} is the mean performance of subgroup k for the baseline model. The local (ΔL) changes are then:

$$\Delta L = \frac{a_p^{new} - a_q^{base}}{a_q^{base}} \tag{5.2}$$

where a_p^{new} is the mean performance of the worst-case subgroup for a chosen debiased model and a_q^{base} is the mean performance of the worst-case subgroup for the baseline model (p and q need not be the same).

5.2.3 True Positive Rate Disparity

We also calculate the true positive rate (TPR), also called sensitivity (Equation 5.3) and TPR disparity. TPR is a good metric to use here as it works well in scenarios with imbalanced datasets. TPR disparity is calculated for binary subgroups as the difference between the TPR of the subgroup value and the TPR of the other subgroup value. For non-binary subgroups it is calculated as the difference between the TPR of the subgroup value and the mean TPR of that subgroup (Equation 5.4).

$$TPR = \frac{TP}{TP + FN} \tag{5.3}$$

$$TPRD_{S_j,i} = TPR_{S_j,i} - Median(TPR_{S_j,1}...TPR_{S_j,k}) \tag{5.4}$$

where $TPR_{S_j,i}$ represents the TPR or the j^{th} attribute subgroup and the i^{th} attribute value in that subgroup.

5.2.4 Fairness Analysis

In addition to using standard evaluation metrics, we can use counterfactuals themselves to estimate the bias in a classifier that predicts a discrete label, by comparing predicted labels for original and counterfactual images. In [37] a classifier is defined to be biased if the predicted label changes for a label-preserving counterfactual image and if it changes the label to one class from another more often than vice versa. This can be expressed as:

$$bias = p(y_r \neq y_c)\,[p(y_r = 0, y_c = 1 | y_r \neq y_c) - p(y_r = 1, y_c = 0 | y_r \neq y_c)] \tag{5.5}$$

which gives a representation of the bias as a number between -1 and 1. In the case when the classifier is unbiased the number will be 0, when it is negative then the classifier is more likely to change the label to 0 and equivalently it is positive if the classifier is more likely to change the label to 1.

We can use this method to estimate classifier bias and express its fairness visually, constructing two-dimensional plots of the predictions for original and augmented images. In the ideal case, where there is no bias present and the classifier can be labelled as “fair”, all the points should be clustered along the $x = y$ line. Clusters of points in the top-left quadrant mean the label was changed from 0 to 1 - meaning there are a lot of false positives - and analogously clusters of points in the bottom-right quadrant mean the label was changes from 1 to 0 - meaning there are false negatives present.

It is also beneficial to compare fairness plots for the difference in predictions for original images and images augmented using both standard augmentation methods and the counterfactual method. With this comparison, we can see how “useful” a given augmentation method is by itself and in comparison with others. Given a debiased classifier with improved performance, we can check its fairness using a chosen augmentation technique. If the results indicate that the classifier is unfair, even though other evaluation metrics prove differently, then this is an indication that the method being analysed is not a valid way of estimating classifier bias, and hence debiasing using that method does not provide much empirical evidence for being beneficial in bias mitigation.

5.2.5 Dimensionality Reduction

To analyse and explore the effects of debiasing datasets with various methods on the representations the model learns, it is useful to look at the learnt embeddings and analyse patterns that emerge. With high-dimensional data it is hard to capture the embeddings visually, therefore dimensionality reduction is commonly used to reduce the dimensions of datapoints and generate 2D plots of data.

T-SNE is a method commonly used for this task, introduced in [46]. It is a highly-adaptable technique which can give an intuition for how the data is arranged or how it correlates internally. T-SNE aims to capture similarity between samples in the high-dimensional feature space. It is often used in tandem with PCA [47] which is another dimensionality reduction technique. The new dimensions obtained after applying PCA represent the largest variation in the high-dimensional feature space. Therefore, the first few dimensions of PCA will contain the strongest separation of data based on the attributes the model was trained on. Because t-SNE is computationally expensive, often times PCA is applied first to reduce the dimensions of the data which is then fed into t-SNE.

Here we use dimensionality reduction for two reasons. Firstly, we samples from the test set and extract embeddings generated by the trained models to see how good they are in clustering test data. Secondly, we can extract embeddings for original data and counterfactually augmented data and thus try to spot bias around different attributes.

5.2.6 Metrics: Implementation

An evaluation pipeline was implemented to evaluate the different methods used in this comparison. We mostly made use of standard python libraries such as *scikit-learn* [48] and *statistics* to get the basic evaluation metrics. To generate comparison plots *matplotlib* [49] and *seaborn* [50] were used. For dimensionality reduction we used ready implementations from *sklearn.decomposition* and *sklearn.manifold* to apply both PCA and t-SNE.

Chapter 6

Results & Analysis

In this chapter, we describe the experiments performed on all previously introduced setups. We present both quantitative and qualitative results and analyse the impact of each tested method.

6.1 MNIST

To evaluate the counterfactual data augmentation method on the Morpho-MNIST and Colored-MNIST datasets, a baseline trained predictive model and four standard debiasing methods were used as reference for the comparison. The methods were oversampling, standard data augmentations (rotations, blurring, adding noise) and mixup for Morpho-MNIST as well as Group DRO for Colored-MNIST.

All hyperparameters were tuned for the baseline models and kept the same for all further debiasing methods.

6.1.1 Quantitative Evaluation

Looking at some standard metrics for Morpho-MNIST we can see that all debiasing methods have an impact on the classifier's performance, improving it as expected. We note an improvement in results over standard debiasing methods for both the counterfactual augmentations expansion and the CF regularisation method. It is clear that the most significant change was observed for the CF regularisation method, both overall (Table 6.1) and per-class (Figure 6.1). There is no loss in recall or precision as well and metrics for all classes are evened out.

Looking at Morpho-MNIST TPR disparity for the thickness attribute, (Figure 6.3) we can see that while the standard augmentation method performs well in improving performance per each class, it also introduces a rather significant TPR disparity between labels and all attribute subgroups. Both the counterfactual and CF regularisation methods remove the difference between thick and thin samples almost completely, with the counterfactual expansion method achieving a lower TPR range

	Accuracy	F1-score	Precison	Recall	ROC-AUC	Bias
Baseline	89.73	89.51	89.84	89.7	99.23	-0.013
Oversampling	93.32	93.2	93.31	93.32	99.64	-0.005
Std Augs	91.31	91.31	80.1	83.32	99.72	-0.017
Mixup	93.12	93	93.34	93.1	99.71	0.0123
CFs (do_t)	94.22	94.1	94.22	94.24	0.997	0.005
CF reg (do_t)	**97.52**	**97.54**	**97.6**	**97.51**	**99.9**	**0.004**

Table 6.1: Morpho-MNIST overall performance comparison. For accuracy, F1-score, precison, recall and ROC-AUC score higher is better and for bias lower is better. Bias is measured using Equation 5.5 for the thickness attribute. For counterfactual data augmentation methods do_t signified an intervention on thickness.

(approx. 0.91) than the CF regularisation method (approx 1.0). We can observe similar behaviour for per-class TPR disparity. We can therefore conclude that the counterfactual methods improve local performance evenly among all subgroups while also increasing global performance.

Similarly, for Colored-MNIST, it is clear that the counterfactual regularisation method achieves the best improvement. Counterfactual dataset expansion performs slightly worse than standard augmentations (Table 6.2). Counterfactual regularisation provides steady improvement among all classes as well as color subgroups with minimal disparity which is in comparison to the baseline methods (Figure 6.2).

As previously mentioned, we also run an experiment with 0% bias-conflicting samples. For this setup we train the DSCM on a fully-biased version of Colored-MNIST as well to simulate a real-life scenario well. As this is the case we expect the quality of generated counterfactuals to be reduced as well which is what we observe. It is however clear that the counterfactual methods are still capable of improving the model's performance. In this scenario they significantly outperform any baseline methods. This is to be expected, as it is impossible to remove the spurious color-digit correlation with standard debiasing.

6.1.2 Qualitative evaluation

To evaluate the improvement in fairness we perform fairness analysis around the thickness attribute for Morpho-MNIST and around the color attribute for Colored-MNIST. We compare predictions of trained models between original samples and images with modified thickness. Evaluating fairness with the previously described method requires a binary classification setup, therefore we test the task of predicting whether a given image is of digit class x or not, rather than testing multilabel digit classification. We sample 10k images, labeled as 9, from the test set and for each generate seven counterfactuals, changing the thickness/color of the digit. Then for

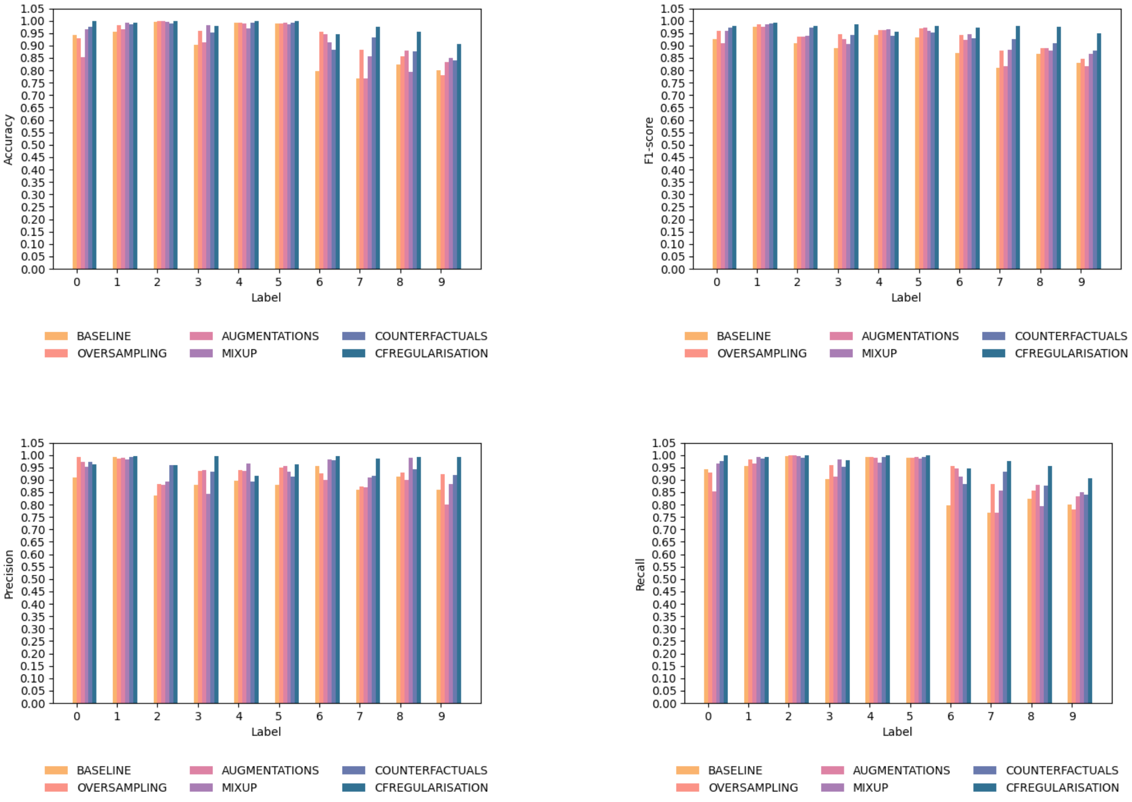

Figure 6.1: Standard metrics for digit classification on Morpho-MNIST. From left to right (top to bottom): accuracy, F1-score, precision, recall

each trained model we compare the predicted probability that a given digit is a 9, as from analysing model performance it is clear the baseline was underperforming for digit 9 with respect to other digits.

For Morpho-MNIST, we observe a significant improvement in bias (Equation 5.5) from -0.0129 for the baseline classifier to 0.0039 for CF regularised model. A comparison of biases for different models debiased using different methods (Figures 6.4, 6.5) shows that the CF regularisation method achieves the highest improvement.

6.2 Mimic CXR

As previously mentioned, we define two tasks on the Mimic CXR dataset, namely binary disease classification and race classification.

Disease Classification

For disease classification we observed a bias with respect to the “race” attribute therefore we chose to target it by balancing the attribute counts using the standard,

	Accuracy	**F1-score**	**Precison**	**Recall**	**ROC-AUC**
		0% bias-conflicting			
Baseline	19.29	20.25	24.58	18.96	57.45
Oversampling	19.41	19.11	19.71	19.98	56.57
StdAugs	12.81	12.71	12.82	12.62	52.25
Group DRO	18.02	17.99	20.1	17.69	55.75
CFs (do_c)	52.86	52.14	54.8	52.54	85.89
CF reg (do_c)	**71.67**	**71.19**	**73.17**	**71.29**	**96.33**
		1% bias-conflicting			
Baseline	73.53	73.6	75.44	73.42	94.27
Oversampling	88.78	88.66	89.49	77.3	98.7
StdAugs	90.83	90.65	91.33	90.82	99.4
Group DRO	60.26	58.43	59.78	59.54	89.57
CFs (do_c)	86.22	86.06	86.45	86.05	98.57
CF reg (do_c)	**93.45**	**93.33**	**93.41**	**93.34**	**99.65**

Table 6.2: Overall performance comparison of digit classification on Colored-MNIST with 1% and 0% bias-conflicting samples. For counterfactual data augmentation methods do_c signifies an intervention on color.

reference methods and the counterfactual method. For counterfactual regularisation, for each batch of original images a random race value is chosen for which counterfactual images are generated for comparison.

Additionally, there is a visible correlation between the patient's age and whether or not they are diagnosed as healthy. Younger patients (0-39 years old) are more likely to be healthy and older patients (80-100 years old) are most often diagnosed with the disease. To counter this imbalance we use regularisation around the age-disease correlation.

Race Classification

For the race classification task we observe a significant drop in performance for Asian and Black patients therefore we decided to balance those subgroups. We also notice that TPR drops for younger patients in comparison to older patients and so we also target the age-disease correlation for this task.

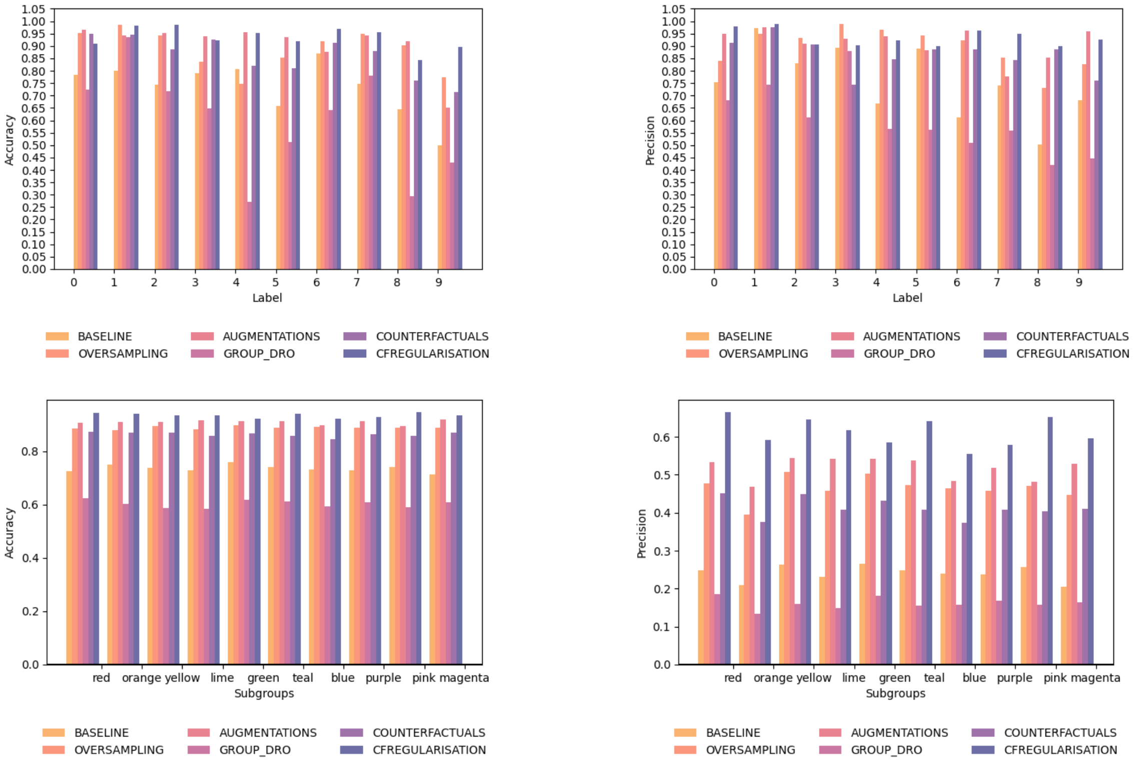

Figure 6.2: Comparison of per-class and per-color attribute subgroup metrics for Colored-MNIST dataset with 1% bias-conflicting samples. From left to right (top to bottom): per-class accuracy, per-class precision, per-subgroup accuracy, per-subgroup precision.

6.2.1 Quantitative Evaluation

Disease Classification

For Mimic CXR we do not observe a significant improvement for the counterfactual methods in comparison with standard techniques. Again, all methods used have an impact on the model, however the counterfactual methods are performing very similarly to data augmentations and oversampling. One interesting thing to notice is the improvement in precision over all race subgroups, which is the best for the CF regularisation method. Conversely, we also notice a drop in recall and no significant improvements overall. This lack of improvement could be caused by the complex correlations between image attributes, where targeting one bias might unintentionally impact another one.

Race Classification

For the race classification task we observe more varied results. While oversampling is counter-effective and decreases performance, both standard data augmentations and mixup make an improvement on precision for Asian and Black samples. Counterfactual data augmentation and CF regularisation also boost per-class precision

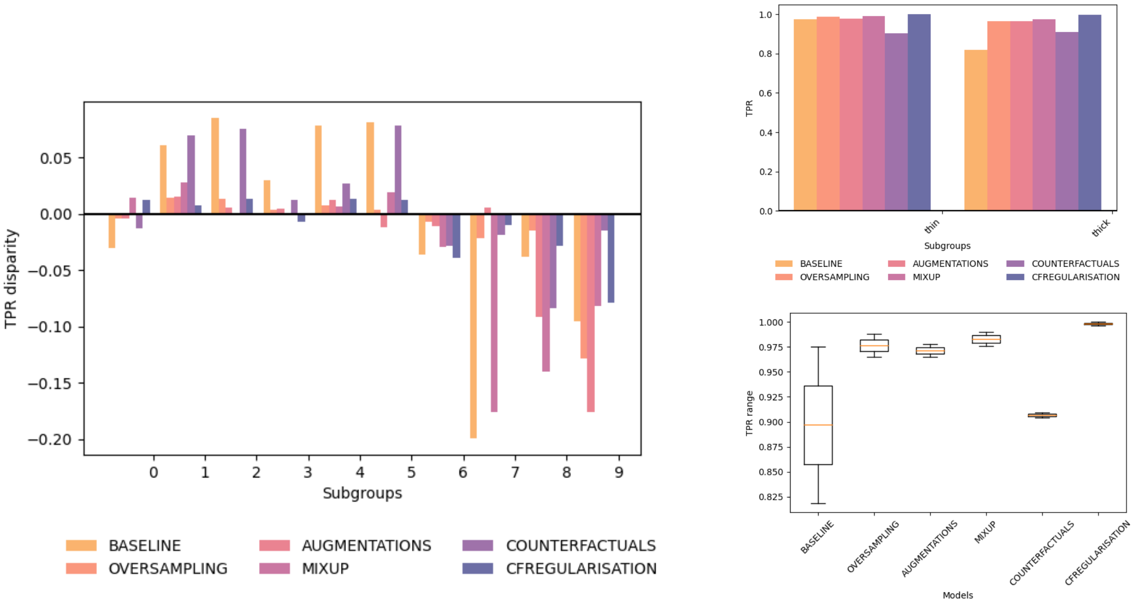

Figure 6.3: TPR disparity per thickness subgroup comparison for the Morpho-MNIST dataset. TPR disparity per class (left), TPR values per thickness range (top-right) and box plot depicting thickness TPR disparity (bottom-right).

however to a lower extent.

Furthermore, all methods improve TPR disparity for all attributes however there is no definitive best result. No method harms per-attribute performance, preserving values similar to the ones achieved for the baseline model.

6.2.2 Qualitative Evaluation

We sample 1k original images from the test set and for each generate seven counterfactual images altering the race attribute to Black to determine the fairness of trained disease classifiers. We use all trained classifiers to predict whether the image is of a healthy patient or not and compare the original and augmented predictions (Figure 6.7). If a classifier is fair and unbiased then all the points should be clustered along the $x = y$ line. For the baseline model we observe both false positives (upper-left quadrant) and false negatives (lower-right quadrant) meaning the model incorrectly classifies healthy patients as unhealthy and vice-versa after the race of the patient is changed to Black. For standard debiasing methods used (oversampling, data augmentations, group DRO) we do not observe much improvement. Similarly, on the plot corresponding to counterfactual data augmentations we still observe false positives and false negatives. The model debiased with CF regularisation proves the most efficient and fair as most of the points are clustered along the $x = y$ line.

We calculate the classifier's fairness and observe a substantial improvement in bias. For disease classification the baseline classifier reports a bias score of -0.018 against race Black and the CF regularised classifier obtains a bias score of 0.003.

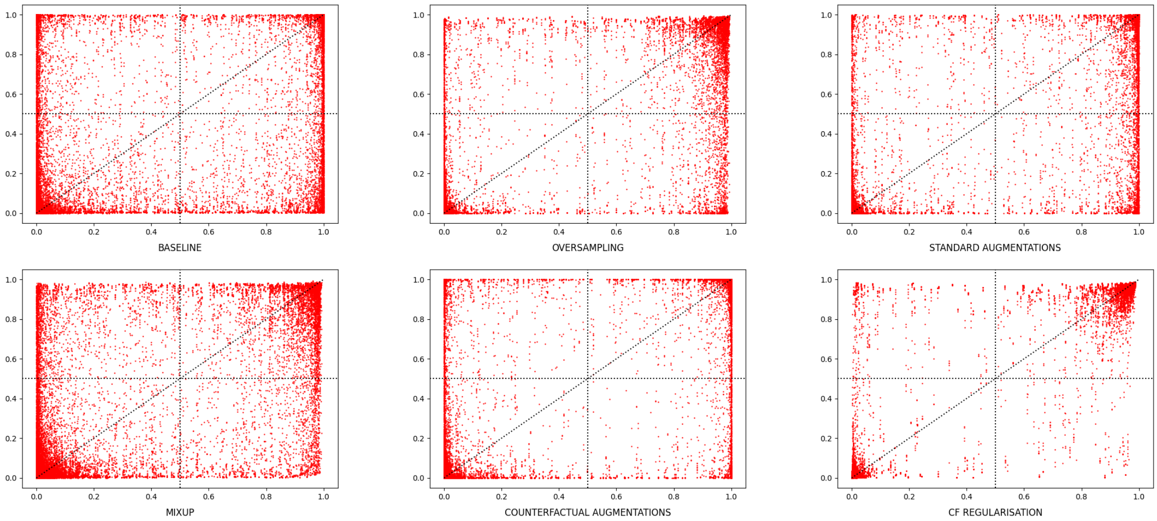

Figure 6.4: Fairness analysis (bias estimation) for digit classification task on Morpho-MNIST. The top left plot is generated using the baseline model with no balancing techniques employed. The remaining plots correspond to predictive models trained using different debiasing methods, from left to right (top to bottom): oversampling, standard data augmentations, mixup, counterfactual data augmentations dataset expansion and CF regularisation. The debiasing methods target the thickness attribute bias.

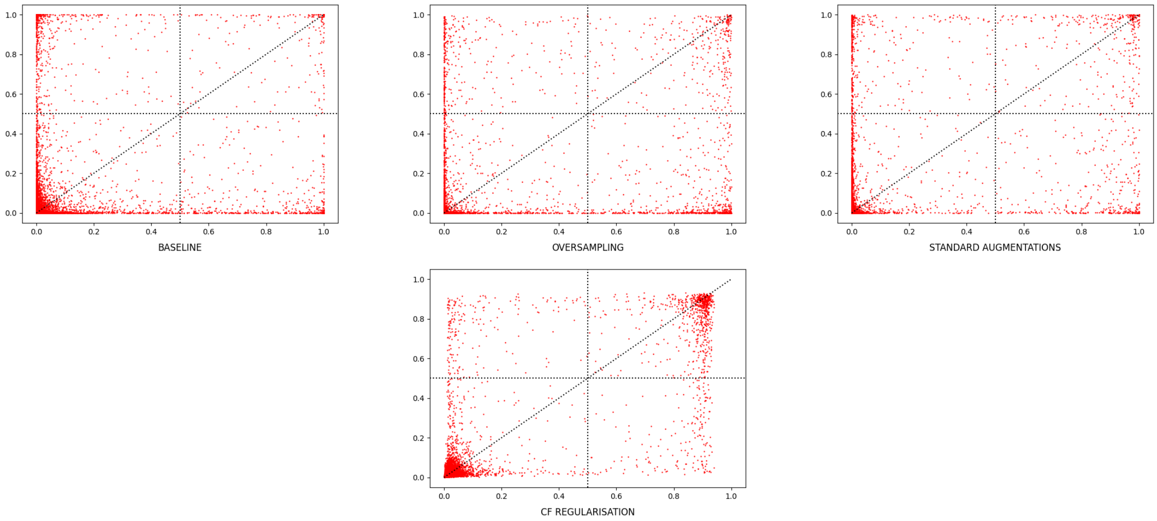

Figure 6.5: Fairness analysis (bias estimation) for digit classification task on Colored-MNIST. The top left plot is generated using the baseline model with no balancing techniques employed. The remaining plots correspond to predictive models trained using different debiasing methods, from left to right (top to bottom): oversampling, standard data augmentations and CF regularisation. The debiasing methods target the color attribute bias.

	Accuracy	**F1-score**	**Precison**	**Recall**	**ROC-AUC**
Baseline	89.58	89.51	89.43	89.63	94.95
Oversampling	88.29	88.18	88.18	88.18	94.1
StdAugs	89.03	88.98	88.89	89.15	94.19
Group DRO	89.27	89.2	89.14	89.27	94.91
CFs (do_r)	89.63	89.56	89.49	89.67	94.85
CFs (do_r) + mixup	89.88	89.81	89.74	89.91	94.97
CFs (rs)	89.88	89.79	89.79	89.77	95.29
CF reg (do_r)	88.9	88.84	88.77	89.04	94.6

Table 6.3: Disease classification overall performance comparison. do_r and rs signify an intervention on race and random sampling, respectively.

	Accuracy	**F1-score**	**Precison**	**Recall**	**ROC-AUC**
Baseline	86.65	69.51	69.94	69.1	88.66
Oversampling	86.09	67.14	70.38	64.65	87.45
Std Augs	88.86	72.22	82.4	67.05	90.56
Mixup	89.7	72.25	**86.1**	**67.08**	89.25
CFs (do_r)	88.17	69.53	73.75	66.56	91.08
CFs (do_r) + mixup	89.11	66.48	78.55	63.94	90.68
CFs (rs)	**90.12**	**76.04**	78.36	74.09	**92.94**
CF reg (do_a, do_f)	88.59	69.38	78.81	64.69	88.86

Table 6.4: Race classification overall performance comparison. For counterfactual data augmentation methods do_r, do_a, do_f and rs signify an intervention on race, age, finding and random sampling, respectively.

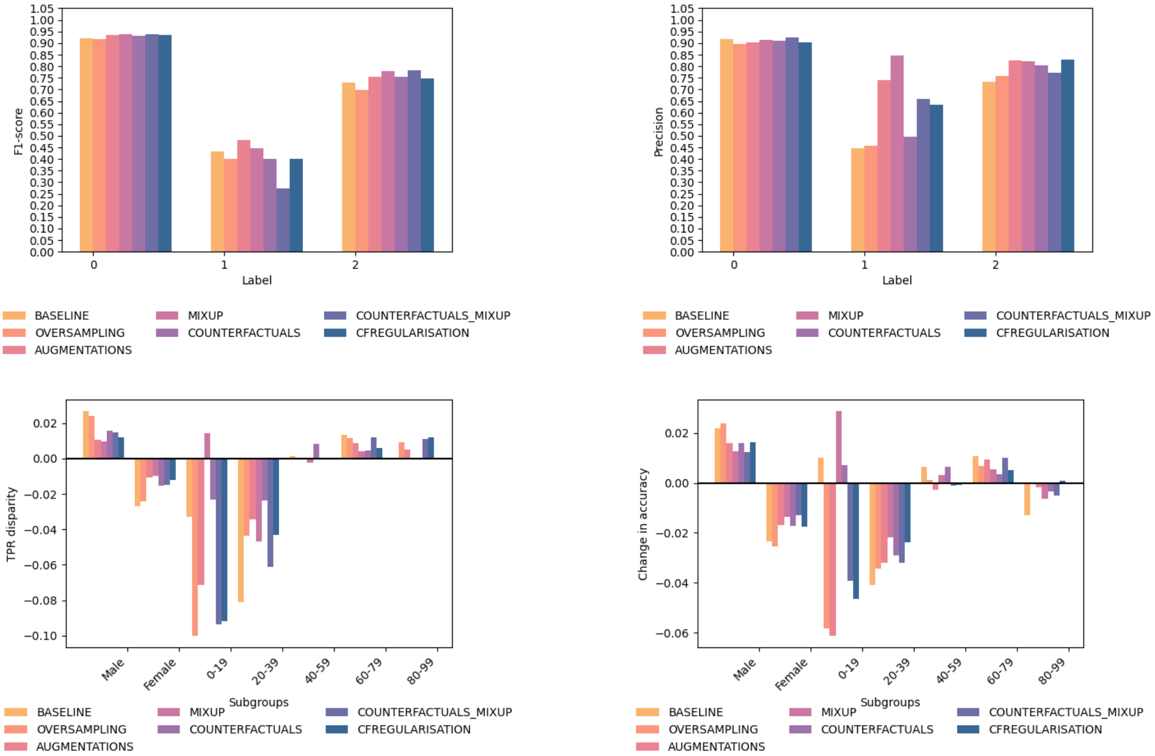

Figure 6.6: Per-class comparison of training results for the race classification task on Mimic CXR. From left to right: per-class F1-score, per-class precision, per-attribute subgroup TPR disparity, per-attribute subgroups change in accuracy with respect to global model accuracy.

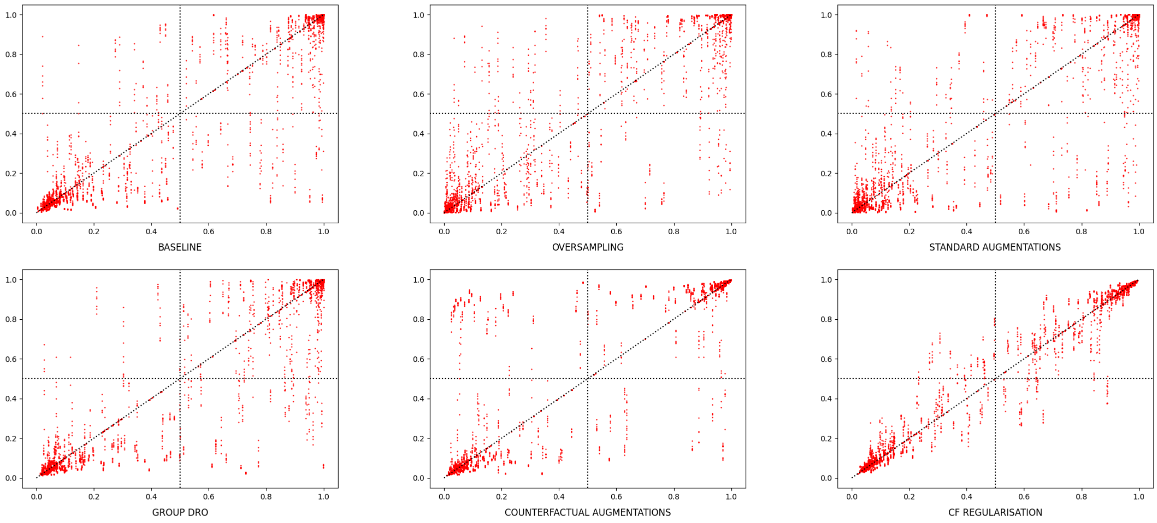

Figure 6.7: Fairness analysis (bias estimation) for disease classification task. The top left plot is generated using the baseline model with no balancing techniques employed. The remaining plots correspond to predictive models trained using different debiasing methods, from left to right (top to bottom): oversampling, standard data augmentations, group DRO, counterfactual data augmentations dataset expansion and CF regularisation. The debiasing methods target the race attribute bias.

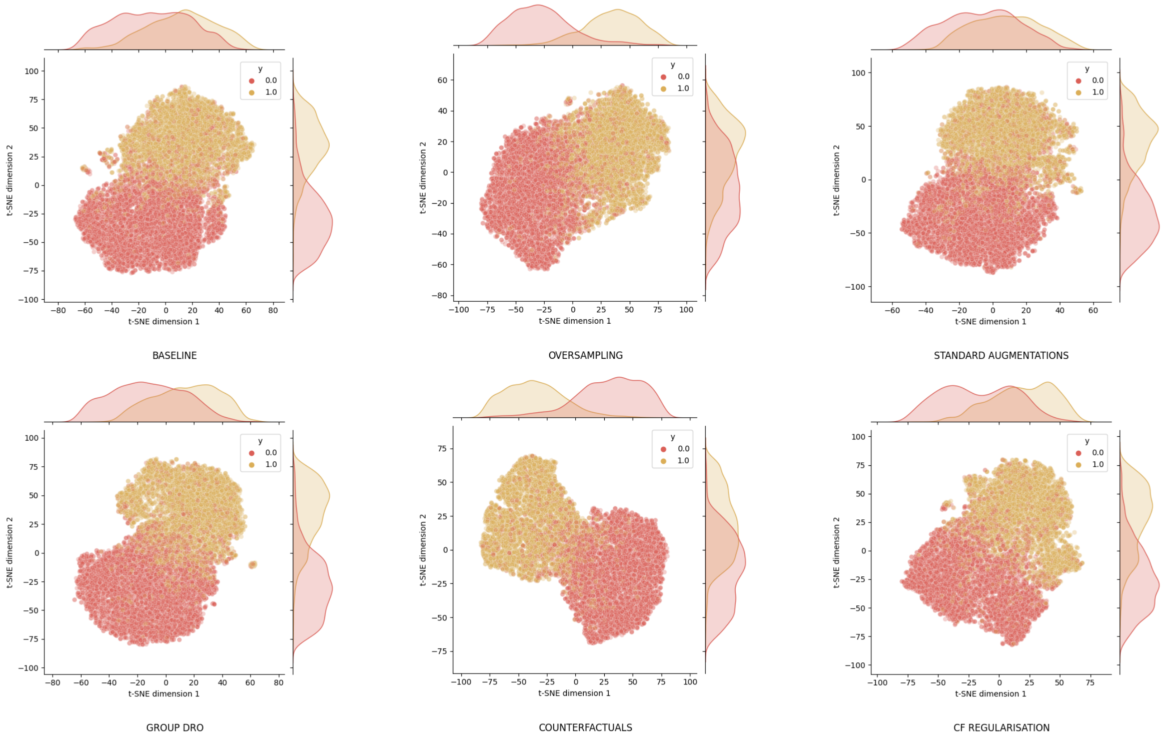

Figure 6.8: T-SNE dimensionality reduction on the test set for disease prediction classifiers. We observe more pronounced clustering and less overlap for the model debiased using the counterfactual expansion method.

6.3 Conclusion

From the results obtained, it is clear that counterfactual data augmentation is a valid debiasing technique which succeeds at targeting certain imbalance present in training datasets. Whether it provides significant improvements over other, more commonly used techniques, seems to depend on the complexity of the setup and correlation biases present.

We have observed how counterfactual data augmentation can improve a predictive model, outperforming widely-used methods, on synthetically created Morpho-MNIST and Colored-MNIST datasets. The important thing to note about the MNIST datasets is that there are only a few attributes describing each image, which we are in full control of. We are also aware of all the relationships between attributes. Further, the test set is perfectly balanced which enables fair evaluation of trained models.

The Mimic CXR dataset, in comparison, is a collection of real-world data. It exhibits multiple types of bias and balancing one of them can, unintentionally, influence another one. Due to data scarcity, the test set in use reflects the imbalance present in the training data, making fair testing a challenge. Further, there are some known issues in the current version on the Mimic CXR DSCM. Namely, intervening on certain attributes such as race can have an impact on others, for instance amplifying the disease present in a given x-ray. Nevertheless, we have observed a substantial improvement in the model's fairness when debiasing with the counterfactual regularisation method.

Taking all above into consideration, we can conclude that counterfactual data augmentation can indeed mitigate bias introduced by spurious correlations in data and can match the performance or even outperform many commonly used methods, both model and data-driven.

Furthermore, in terms of explainability, counterfactual data augmentation is especially useful as all augmentations are fully based on causal relationships which is often not the case for more commonly used techniques.

We also acknowledge that there is still opportunity for improvement in the current implementation of the generative model used in this evaluation – the DSCM. It is highly likely, that a refined model would further improve performance both on synthetic and real-world data.

Chapter 7

Discussion

In this chapter, we outline a few potential areas of future work and describe the broader research context which this project resides in (Section 7.1).

We also cover any ethical issues taken into consideration for this project (Section 7.2).

7.1 Outlook & Future Work

7.1.1 Other Computer Vision Tasks

The evaluation in this project focuses solely on image classification as it is most intuitive to apply counterfactual inference for this task. It would, however, be interesting to see counterfactual data augmentation applied in other computer vision tasks such as regression or structured prediction (e.g image segmentation).

7.1.2 Reducing Data Analysis

Performing bias analysis is essential to use any data augmentation technique efficiently. Similarly, it is crucial to understand the spurious correlations between attributes present in training data to correctly and efficiently apply counterfactual inference. It is, however, a time-consuming activity, and, ideally, we would like to reduce it to a minimum while keeping the efficacy of the method. An attempt at enabling this is the random sampling experiments we ran, as those do not require modifications tailored to specific biases. We observed an improvement for randomised debiasing compared to the more targeted mitigation.

7.1.3 Counterfactuals Evaluation and Uncertainty Analysis

As part of this project we conduct fairness analysis of the trained classifiers. We do not, however, evaluate the generated counterfactuals as we make the assumption that it might not be of necessity for bias mitigation. It would be, nonetheless, interesting to perform further evaluation and measure the uncertainty of generated counterfactual images. Quantification of uncertainty in counterfactual inference would allow for identifying flaws and limitations in the generation process. These could then be used for implementing potential improvements.

7.2 Ethical Considerations

7.2.1 Protection of Personal Data

This research is exempt from ethical approval as the analysis is based on secondary data which is publicly available, and no permission is required to access the data.

Furthermore, all datasets used in this project are open source. Multiple data augmentation methods used for generating both augmented data points and counterfactuals involve modifying the image data. All of the datasets are available under the Creative Commons (CC) License which allows distribution, remix, adaptation, and building upon the material in any medium or format, so long as attribution is given to the creator. Moreover, all data available as part of the datasets is anonymized. For datasets which include sensitive data, accepted and well-established methodologies will be followed.

7.2.2 Potential Misuse

Counterfactuals have a potential for being misused in some contexts. For example, generating plausible counterfactual images could be misused in a medical context to trick a machine learning model. As previously mentioned, there are more and more medical systems, especially classification systems, using machine learning to improve the speed and accuracy of processes usually performed by humans. Lets consider a situation where government medical funding is distributed based on some classification process which is done by analysing medical images submitted from patients. A medical facility could generate plausible counterfactual images and use those to receive additional government funding if those decision processes are done using machine learning techniques. This project does not directly aim to develop tools for generating image counterfactuals but rather evaluate how efficient using counterfactuals can be in improving performance of neural networks. The results from model evaluation however could potentially be used as guidelines on how to best generate counterfactual images to fit the classification criteria of a specific neural network.

We should also consider misuse from the perspective of human users. Machine learning systems can be used malevolently to trick human observers. One example of such use is DeepFakes which aim to generate images or videos where they replace the likeness of one person with another and can be used to generate fake and misleading content. Counterfactuals can be used in a similar way by altering input images and generating fake, possibly harmful outputs.

Bibliography

[1] Yann LeCun, Yoshua Bengio, and Geoffrey Hinton. Deep learning. *Nature*, 521 (7553):436–444, 2015. pages 1, 5

[2] William Gale, Luke Oakden-Rayner, Gustavo Carneiro, Andrew P Bradley, and Lyle J Palmer. Detecting hip fractures with radiologist-level performance using deep neural networks. *arXiv preprint arXiv:1711.06504*, 2017. pages 1

[3] Paras Lakhani and Baskaran Sundaram. Deep learning at chest radiography: automated classification of pulmonary tuberculosis by using convolutional neural networks. *Radiology*, 284(2):574–582, 2017. pages 1

[4] Scott Mayer McKinney, Marcin Sieniek, Varun Godbole, Jonathan Godwin, Natasha Antropova, Hutan Ashrafian, Trevor Back, Mary Chesus, Greg S Corrado, Ara Darzi, et al. International evaluation of an ai system for breast cancer screening. *Nature*, 577(7788):89–94, 2020. pages 1

[5] Jungsoo Lee, Eungyeup Kim, Juyoung Lee, Jihyeon Lee, and Jaegul Choo. Learning debiased representation via disentangled feature augmentation. *Advances in Neural Information Processing Systems*, 34:25123–25133, 2021. pages 1, 10

[6] Maximilian Ilse, Jakub M Tomczak, and Patrick Forré. Selecting data augmentation for simulating interventions. In *International Conference on Machine Learning*, pages 4555–4562. PMLR, 2021. pages 1, 8, 10

[7] Tian Xia, Pedro Sanchez, Chen Qin, and Sotirios A Tsaftaris. Adversarial counterfactual augmentation: Application in alzheimer's disease classification. *arXiv preprint arXiv:2203.07815*, 2022. pages 2

[8] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016. pages 5, 33

[9] Xavier Glorot and Yoshua Bengio. Understanding the difficulty of training deep feedforward neural networks. In *Proceedings of the thirteenth international conference on artificial intelligence and statistics*, pages 249–256. JMLR Workshop and Conference Proceedings, 2010. pages 5

[10] Aston Zhang, Zachary C. Lipton, Mu Li, and Alexander J. Smola. Dive into deep learning. *arXiv preprint arXiv:2106.11342*, 2021. pages 5, 6

[11] Saining Xie, Ross Girshick, Piotr Dollár, Zhuowen Tu, and Kaiming He. Aggregated residual transformations for deep neural networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1492–1500, 2017. pages 6

[12] Sergey Zagoruyko and Nikos Komodakis. Wide residual networks. *arXiv preprint arXiv:1605.07146*, 2016. pages 6

[13] Gao Huang, Zhuang Liu, Laurens van der Maaten, and Kilian Q. Weinberger. Densely connected convolutional networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, July 2017. pages 6

[14] Joy Buolamwini and Timnit Gebru. Gender shades: Intersectional accuracy disparities in commercial gender classification. *Conference on fairness, accountability and transparency*, pages 77–91, 2018. pages 7

[15] Lisa Anne Hendricks, Kaylee Burns, Kate Saenko, Trevor Darrell, and Anna Rohrbach. Women also snowboard: Overcoming bias in captioning models. In *Proceedings of the European conference on computer vision (ECCV)*, pages 771–787, 2018. pages 7

[16] Tolga Bolukbasi, Kai-Wei Chang, James Y Zou, Venkatesh Saligrama, and Adam T Kalai. Man is to computer programmer as woman is to homemaker? debiasing word embeddings. *Advances in neural information processing systems*, 29, 2016. pages 7

[17] Jieyu Zhao, Tianlu Wang, Mark Yatskar, Vicente Ordonez, and Kai-Wei Chang. Men also like shopping: Reducing gender bias amplification using corpus-level constraints. *Pro- ceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, 2017. pages 7

[18] Nick Pawlowski, Daniel Coelho de Castro, and Ben Glocker. Deep structural causal models for tractable counterfactual inference. *Advances in Neural Information Processing Systems*, 33:857–869, 2020. pages 7, 13, 14

[19] Laleh Seyyed-Kalantari, Guanxiong Liu, Matthew McDermott, Irene Y Chen, and Marzyeh Ghassemi. Chexclusion: Fairness gaps in deep chest x-ray classifiers. In *BIOCOMPUTING 2021: proceedings of the Pacific symposium*, pages 232–243. World Scientific, 2020. pages 7, 33

[20] Ben Glocker, Charles Jones, Mélanie Bernhardt, and Stefan Winzeck. Algorithmic encoding of protected characteristics in chest x-ray disease detection models. *Ebiomedicine*, 89, 2023. pages 7

[21] Yann LeCun and Corinna Cortes. The mnist database of handwritten digits. 2003. URL `http://yann.lecun.com/exdb/mnist/`. pages 7, 13

[22] Terrance DeVries and Graham W Taylor. Improved regularization of convolutional neural networks with cutout. *arXiv preprint arXiv:1708.04552*, 2017. pages 7, 8, 9, 11

[23] Daniel C Castro, Ian Walker, and Ben Glocker. Causality matters in medical imaging. *Nature Communications*, 11(1):1–10, 2020. pages 8, 9

[24] Hongyi Zhang, Moustapha Cisse, Yann N Dauphin, and David Lopez-Paz. mixup: Beyond empirical risk minimization. *arXiv preprint arXiv:1710.09412*, 2017. pages 8, 10, 20

[25] Evan Z Liu, Behzad Haghgoo, Annie S Chen, Aditi Raghunathan, Pang Wei Koh, Shiori Sagawa, Percy Liang, and Chelsea Finn. Just train twice: Improving group robustness without training group information. In *International Conference on Machine Learning*, pages 6781–6792. PMLR, 2021. pages 8

[26] Shiori Sagawa, Pang Wei Koh, Tatsunori B Hashimoto, and Percy Liang. Distributionally robust neural networks for group shifts: On the importance of regularization for worst-case generalization. *arXiv preprint arXiv:1911.08731*, 2019. pages 8, 10, 20, 22

[27] Byungju Kim, Hyunwoo Kim, Kyungsu Kim, Sungjin Kim, and Junmo Kim. Learning not to learn: Training deep neural networks with biased data. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9012–9020, 2019. pages 8

[28] Jing An, Lexing Ying, and Yuhua Zhu. Why resampling outperforms reweighting for correcting sampling bias with stochastic gradients. *arXiv preprint arXiv:2009.13447*, 2020. pages 10

[29] Judea Pearl. *Causality*. Cambridge university press, 2009. pages 11

[30] Daniel Coelho de Castro and Ben Glocker. Morpho-mnist, 2021. URL `https://github.com/dccastro/Morpho-MNIST`. pages 13, 25

[31] Miguel Monteiro, Fabio De Sousa Ribeiro, Nick Pawlowski, Daniel C Castro, and Ben Glocker. Measuring axiomatic soundness of counterfactual image models. *arXiv preprint arXiv:2303.01274*, 2023. pages 15

[32] Nitesh V. Chawla. *Data Mining for Imbalanced Datasets: An Overview*, pages 853–867. Springer US, Boston, MA, 2005. ISBN 978-0-387-25465-4. doi: 10.1007/0-387-25465-X_40. URL `https://doi.org/10.1007/0-387-25465-X_40`. pages 16

[33] Andrew P Bradley. The use of the area under the roc curve in the evaluation of machine learning algorithms. *Pattern recognition*, 30(7):1145–1159, 1997. pages 16

[34] Jacob Cohen. A coefficient of agreement for nominal scales. *Educational and psychological measurement*, 20(1):37–46, 1960. pages 16

[35] Divyansh Kaushik, Eduard Hovy, and Zachary C Lipton. Learning the difference that makes a difference with counterfactually-augmented data. *arXiv preprint arXiv:1909.12434*, 2019. pages 17

[36] Ananth Balashankar, Xuezhi Wang, Ben Packer, Nithum Thain, Ed Chi, and Alex Beutel. Can we improve model robustness through secondary attribute counterfactuals? In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 4701–4712, 2021. pages 17

[37] Saloni Dash, Vineeth N Balasubramanian, and Amit Sharma. Evaluating and mitigating bias in image classifiers: A causal perspective using counterfactuals. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 915–924, 2022. pages 17, 21, 34

[38] Ziwei Liu, Ping Luo, Xiaogang Wang, and Xiaoou Tang. Deep learning face attributes in the wild. In *Proceedings of International Conference on Computer Vision (ICCV)*, December 2015. pages 17

[39] Guilherme Pombo, Robert Gray, M Jorge Cardoso, Sebastien Ourselin, Geraint Rees, John Ashburner, and Parashkev Nachev. Equitable modelling of brain imaging by counterfactual augmentation with morphologically constrained 3d deep generative models. *Medical Image Analysis*, 84:102723, 2023. pages 17

[40] Abbavaram Gowtham Reddy, Saketh Bachu, Saloni Dash, Charchit Sharma, Amit Sharma, and Vineeth N Balasubramanian. Rethinking counterfactual data augmentation under confounding. *arXiv preprint arXiv:2305.18183*, 2023. pages 17

[41] Jungseock Joo and Kimmo Kärkkäinen. Gender slopes: Counterfactual fairness for computer vision models by attribute manipulation. In *Proceedings of the 2nd international workshop on fairness, accountability, transparency and ethics in multimedia*, pages 1–5, 2020. pages 18

[42] TorchVision maintainers and contributors. Torchvision: Pytorch's computer vision library, 2016. URL `https://github.com/pytorch/vision`. pages 22

[43] Alex Clark. Pillow (pil fork) documentation, 2015. URL `https://buildmedia.readthedocs.org/media/pdf/pillow/latest/pillow.pdf`. pages 22

[44] Alistair EW Johnson, Tom J Pollard, Seth J Berkowitz, Nathaniel R Greenbaum, Matthew P Lungren, Chih-ying Deng, Roger G Mark, and Steven Horng. Mimic-cxr, a de-identified publicly available database of chest radiographs with free-text reports. *Scientific data*, 6(1):317, 2019. pages 26

[45] Robert Carruthers, Isabel Straw, James K Ruffle, Daniel Herron, Amy Nelson, Danilo Bzdok, Delmiro Fernandez-Reyes, Geraint Rees, and Parashkev Nachev. Representational ethical model calibration. *npj Digital Medicine*, 5(1):170, 2022. pages 33

[46] Laurens Van der Maaten and Geoffrey Hinton. Visualizing data using t-sne. *Journal of machine learning research*, 9(11), 2008. pages 35

[47] Karl Pearson. Liii. on lines and planes of closest fit to systems of points in space. *The London, Edinburgh, and Dublin philosophical magazine and journal of science*, 2(11):559–572, 1901. pages 35

[48] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830, 2011. pages 36

[49] J. D. Hunter. Matplotlib: A 2d graphics environment. *Computing in Science & Engineering*, 9(3):90–95, 2007. doi: 10.1109/MCSE.2007.55. pages 36

[50] Michael L. Waskom. seaborn: statistical data visualization. *Journal of Open Source Software*, 6(60):3021, 2021. doi: 10.21105/joss.03021. URL `https://doi.org/10.21105/joss.03021`. pages 36