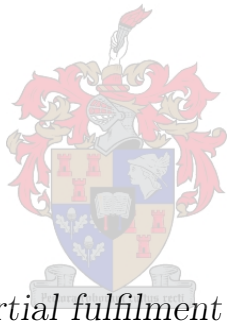
The Class Imbalance Problem In Computer Vision

by

Willem Hendrik Crous



Thesis presented in partial fulfilment of the requirements for the degree of Master of Science (Applied Mathematics) in the Faculty of Science at Stellenbosch University

Supervisor: Prof. W. Brink

April 2022

Declaration

By submitting this thesis electronically, I declare that the entirety of the work contained therein is my own, original work, that I am the sole author thereof (save to the extent explicitly otherwise stated), that reproduction and publication thereof by Stellenbosch University will not infringe any third party rights and that I have not previously in its entirety or in part submitted it for obtaining any qualification.

Date: April 2022

. . . . . .

Abstract

Class imbalance is a naturally occurring phenomenon, typically characterised as a dataset consisting of classes with varying numbers of samples. When trained on class imbalanced data, networks tend to favour frequently occurring (majority) classes over the less frequent (minority) classes. This poses challenges for tasks reliant upon accurate recognition of the less frequent classes. The aim of this thesis is to investigate general methods towards addressing this problem. First we establish why a network may favour majority classes. We contend that as less frequent classes are likely to under-represent the required underlying distribution for a given task, training may produce a decision boundary that transgresses the feature space of minority classes. Additionally we find that the weight norms of the classification layer in a neural network may tend towards the distribution of the training data, thus affecting the decision boundary. We determine that this decision boundary shift impacts both the accuracy and confidence calibration of neural networks. We investigate several approaches to shift the decision boundary. The first approach acquires additional data and increases the representation of minority classes. This is achieved through either creating synthetic samples following a distribution-aware regularisation method, or utilising additional unlabelled data in a semi-supervised setting. The second approach aims to adjust the classifier weight norms by separately training the classifier and feature extractor. We find that implementing an effective regularisation method with a simple decoupled sampling scheme can provide considerable improvements over standard sampling methods. Furthermore we find that utilising additional unlabelled data may lead to additional gains given certain dataset characteristics are taken into consideration.

Acknowledgements

It has been a privilege to undertake further academic studies and I am grateful for the countless inquisitive endeavours this thesis is derived from. Reflecting upon the journey that has lead me to this point in my life, I would like to express my sincere gratitude to the following people and organisations.

- My supervisor Prof Willie Brink, for his patience, understanding and support. His methodological approach, clear minded remarks and guidance has allowed me to deepen my understanding of my academic undertaking. His input has undoubtedly enabled me to produce a more refined and superior piece of work.
- The Faculty of Science at Stellenbosch University, for the years of support and facilitating a culture of excellence for students to strive towards.
- To my close family and friends, for which their acceptance and understanding remains, as always, invaluable. I will forever remain in debt to their tenacious love and support.

Contents

Chapter 1

Introduction

The class imbalance problem is concerned with the performance of machine learning models in the presence of highly skewed (imbalanced) class frequencies. Models such as traditional classifiers (Japkowicz and Stephen, 2002), multi-layer perceptrons (Mazurowski *et al.*, 2008) and convolutional neural networks (CNNs) (Buda *et al.*, 2018) have shown severe performance degradation when trained on class imbalanced datasets. This degradation can be characterised by convergence difficulties during the training phase, as well as generalisation issues on test data where the model might be very bad at recognising the less frequent (minority) classes compared to the most frequent (majority) classes.

1.1 Class imbalanced data

The success of deep neural networks in machine learning application domains such as computer vision, natural language processing and speech recognition has resulted in CNNs largely replacing classical machine learning techniques as the state-of-the-art (Gu *et al.*, 2018). In order to achieve this status, considerable amounts data is required for model training. Consequently, substantial effort has been made towards constructing ever larger, properly annotated datasets.

Ensuring that the accompanying datasets are well annotated and balanced, however, presents new challenges as large scale data collection is likely to lead to class imbalance for which rectification can be impractical or impossible (Longadge and Dongre, 2013). A dataset is said to be class imbalanced when the sample frequencies between classes are not uniformly distributed, but highly skewed.

Class imbalanced datasets can take various forms, although as modern applications increasingly consider larger numbers of classes, large scale datasets

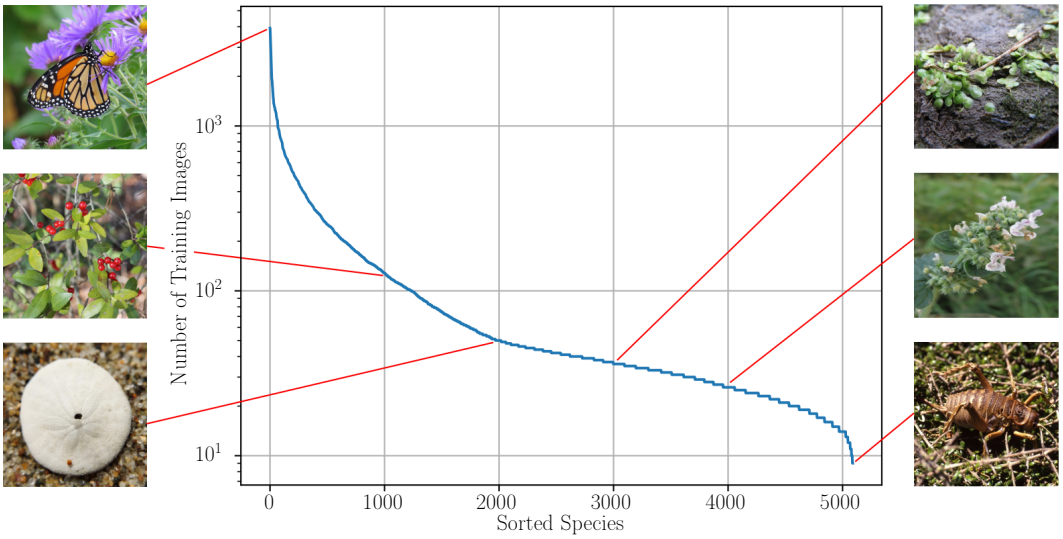


Figure 1.1: The classes of the iNat2017 dataset sorted in descending order according to the number of training images for each class. The dataset exhibits a class imbalanced distribution where the top 1% most populated classes contain over 16% of the total number of training images. Source: Horn *et al.* (2017).

should naturally be governed by some power law, exhibiting a long-tailed (zipf-like) distribution (Horn and Perona, 2017). As an example, Figure 1.1 displays the frequency distribution of samples for each of the roughly 5,000 species in the iNat2017 database. iNat2017 is a dataset constructed from observations of fauna and flora uploaded by citizen scientists across the world (Horn *et al.*, 2017). The dataset clearly exhibits a case where the class distribution is highly imbalanced (notice the log scale on the vertical axis), as just like the real world, some species are disproportionately more likely than others to be observed.

The class distribution of iNat2017 is often referred to as a long-tailed distribution. Long-tailed distributions are characterised by a high frequency of samples from a few classes, followed by a low frequency that gradually tails off asymptotically. Classes at the far end of the tail will have a very low probability of occurrence. Long-tailed distributions are a form of power-law distributions, and encompass important behaviours throughout naturally occurring as well as man-made phenomena (Newman, 2005). Examples of behaviours that exhibit a long-tailed distribution are the occurrence of certain words in a given language, individual wealth distribution, and the intensity of recorded earthquakes. As such, class imbalanced datasets that reflect some form of power-law distribution will be considered for experimentation in this study.

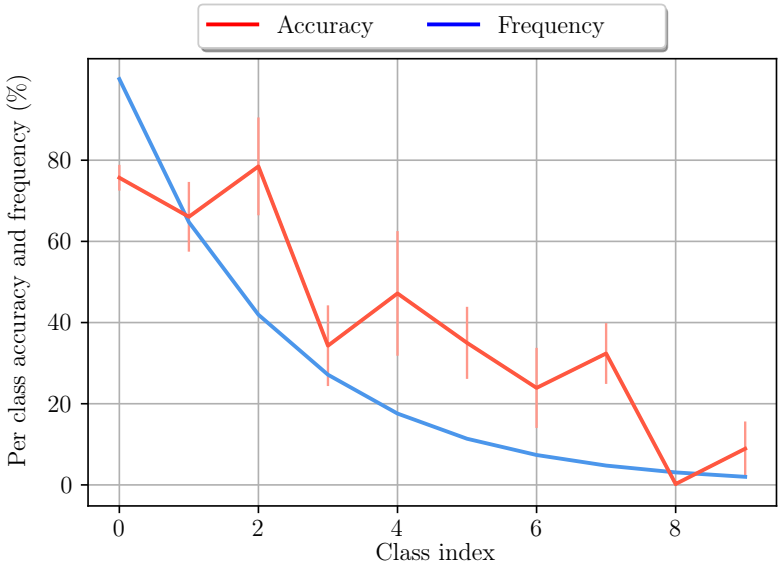


Figure 1.2: Red: averaged per-class accuracy of a standard ResNet-34 network trained on a class imbalanced subset of CIFAR-10, across multiple sessions. Blue: corresponding class frequencies normalised and arranged in descending order. Standard deviation is displayed as bar lines. Class performance is positively correlated with class frequency.

1.2 The class imbalance problem

When the training dataset contains significantly different class frequencies, the network may favour or bias towards classes that occur more frequently. For tasks dependent upon the network's performance on less frequent classes, this may be detrimental.

To demonstrate, Figure 1.2 reports the per-class accuracy of a standard deep neural network trained on an artificially imbalanced dataset with 10 classes. For this particular dataset, the largest class contains nearly 4,000 samples, while the smallest contains only 80 samples. As seen, the measured performance for each class is roughly correlated with class frequency, where majority classes achieve a higher accuracy, indicating that the network has learned to favour majority classes. Dataset and experimental details for experiments like this one will be given in Section 2.2.

This observed bias may be particularly destructive provided the testing criterion for the network is reliant upon the performance on minority classes. Examples of such criteria include the minimum accuracy among all classes, or the accuracy on balanced test distributions, and are common in various practical applications such as fairness or anomaly detection tasks (Branco *et al.*, 2015). For classification tasks, the network's recognition performance on all

classes, if not exclusively the minority, is typically of importance.

An additional challenge when learning from class imbalanced datasets is to reliably estimate how effective the model fits the training data. Standard evaluation criteria are sensitive to skewed distributions and may fail to capture the network's performance on minority classes (Branco *et al.*, 2015). Consequently, decision makers may form misleading conclusions that are not representative of the model's ability. As an example, accuracy is one of the most widely used evaluation metrics to summarise the performance of classification models, and is calculated as the ratio of the number of correctly classified samples to the total number of samples presented. For class imbalanced data, the network may produce a misleadingly high accuracy score by simply classifying samples from the majority classes correctly, without consideration for minority classes.

Furthermore, for evaluating the performance of networks, separate validation and test datasets are typically used. These datasets can be seen as the representative capacity of the required scope of the task that the model needs to perform. However, as they may derive from the same dataset or domain as the training data, they may be class imbalanced as well. Measuring the effectiveness of trained networks on these datasets may as a result introduce additional difficulties where the test data does not contain a sufficient number of representative samples to describe the underlying distribution of minority classes. Consequently, evaluating on these datasets may give misleading results and additional data gathering may be required.

1.3 Background

Horn and Perona (2017) investigate learning from long-tailed distributions. Their findings suggest that the number of samples per class is crucial, demonstrating that performance improves considerably when classes are sufficiently represented (typically having thousands of samples). Conversely, the classification accuracy halves when the number of training samples is reduced by a factor of 10. This implies that as the number of samples for a class decreases, the underlying distribution of the class is diminished, which in turn affects the network's learning capability.

The behaviour seen in Figure 1.2 could therefore be explained as a scenario where the network produces a decision boundary biased towards minority classes. Under the cluster assumption (Rigollet, 2007), a learned decision boundary will not transition through high density regions, or clusters, of features in latent feature space, but will instead optimise towards low density

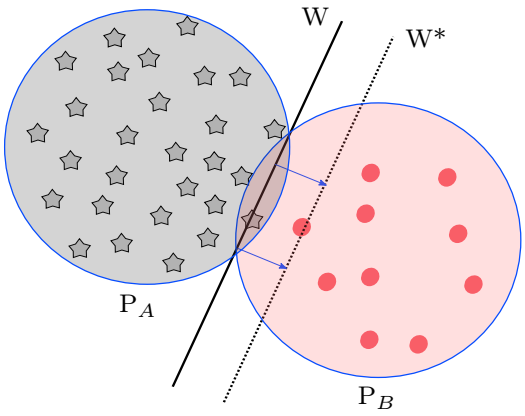


Figure 1.3: Illustration of decision boundary shift when classes consist of disproportionate amounts of representative data. W is the true optimal decision boundary. Unbalanced empirical distributions may lead to a skewed decision boundary W^* that is biased towards the minority class.

regions. Latent features are the learned representations produced by the neural network to distinguish between classes. Points in feature space that are closer to one another are assumed to be similar and represent regions for a particular class. As such, learned decision boundaries should reside in low density regions between these clusters.

In general, a loss function on empirical data is incapable of capturing the actual hidden class distributions. Effectively describing this underlying distribution would require sufficiently representative empirical data, but this is generally not possible for classes with only a few samples (Horn and Perona, 2017). For datasets with varying degrees of representations, typical of imbalanced datasets, learning algorithms may produce decision boundaries which transgress regions of the feature space that are occupied by minority classes.

To demonstrate, we consider a binary toy dataset where one class is underrepresented, shown in Figure 1.3. For this illustration, the shaded regions P_A and P_B represent two underlying class distributions populated with a disproportionate number of samples in latent feature space. For balanced data with samples of both classes accurately describing the underlying distribution, the optimal decision boundary W would successfully separate, while for the given data, the decision boundary W^* would fall further within P_B. Consequently, at test time points belonging to the minority class P_B may be misclassified as P_A, damaging the generalisation of the network. Generalisation refers to a model's relative performance on data not seen during training; the performance when evaluated on previously seen data (training data) compared to data it has never seen before (test data).

As a result, the decision boundary might favour well-represented classes over under-represented classes. To verify experimentally, Figure 1.4 shows the per-class split mean accuracy of a standard CNN trained for a 10-class classification task, using varying levels of class imbalanced training data. The extent of class imbalance is measured by the largest class frequency divided by the smallest class frequency, and is indicated as ρ. Higher values of ρ indicate a more severe level of imbalance. Further details of this construction are given in Section 2.2.

Reporting network performance in terms of the per-class accuracy can be cumbersome and convoluted, so instead it is common to report the accuracy among different frequency splits for efficient and simplistic comparisons (Tang *et al.*, 2020; Zhang *et al.*, 2019; Kang *et al.*, 2020). The splits are constructed by grouping classes into four different groups. The first split contains all classes, denoted as the "all" split. The top third most frequent classes are grouped into the "many" split, and the bottom third least frequent classes in the "few" split. The remaining classes are included in the "medium" split.

As seen in Figure 1.4, learning from class imbalanced data can lead to poor generalisation of less frequent classes, where performance on the few and medium splits progressively deteriorates as the level of class imbalance increases. Note, however, that performance on the many split improves over the balanced case even though the total number of samples in the split decreases. This suggests that the decision boundary becomes more biased towards the minority class region in feature space, with relatively larger numbers of samples available for majority classes. However, it is unclear whether the decision boundary is biased purely as a result of the disproportionate amount of representative data, or if learning from class imbalanced data also affects how neural networks form decision boundaries.

Gathering sufficiently many samples may prove impractical, and to observe a single sample of a minority class may require the observation of a disproportionate number of samples from majority classes. This may result in an exponential cost to dataset construction for improving performance on the minority classes. For example, eBird is a dataset of observations from citizen scientists, and follows a long-tailed distribution (Wood *et al.*, 2011). The dataset contains approximately 550,000 images of 2,215 different classes of bird species. Collecting all of that data required over a year of concerted effort from thousands of participants. Increasing the dataset with additional observations such that the 2,000 most observed classes contain at least 10,000 images would take approximately 100 years (Horn and Perona, 2017).

Although training on larger and more class balanced datasets have shown to improve network performance, our work focuses on methods to leverage

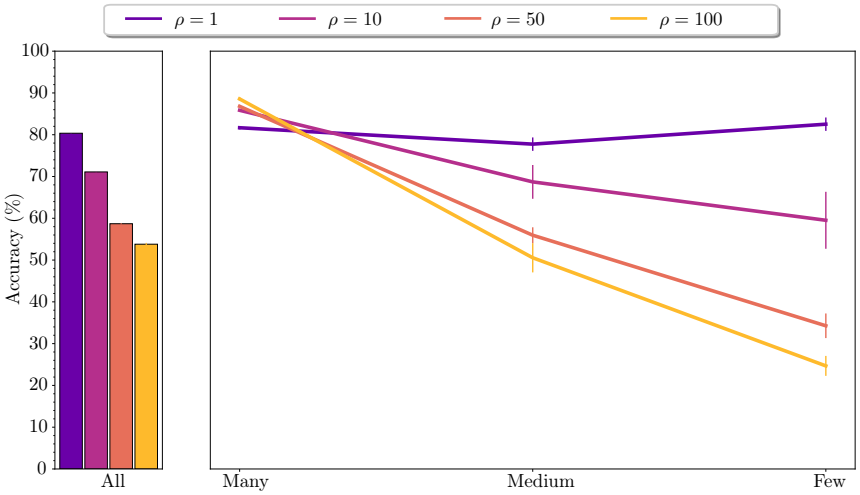


Figure 1.4: Averaged split accuracies for various levels of class imbalance. Classes are split into groups according to sample frequency. Performance on the all split is misleading of the overall performance of the network. Performance on classes in the many split increases as the number of samples in the medium and few splits decreases. Bar lines indicate standard deviation across multiple runs.

available data in scenarios where collecting and annotating additional data is not feasible. Additionally, we will more closely investigate how learning from class imbalanced data affects the decision boundary as well as other important metrics used in machine learning applications.

1.4 Previous work

Various methods have been developed to combat the adverse effects of the class imbalance problem in machine learning (Leevy *et al.*, 2018). In the case of deep neural networks, methods can be grouped into three categories: external methods, internal methods, and hybrid approaches (Krawczyk, 2016). External methods aim to reduce the severity of the imbalance through either leveraging the existing data with data sampling methods, or introducing additional data for training. Internal methods modify the learning procedure to adjust the outputs or predictions of the network and reduce the bias towards majority samples. Hybrid approaches combine sampling and algorithmic methods.

Throughout subsequent chapters, mention of noteworthy methods relating to these categories will be given. Our work is predominantly inspired by two recently published papers, discussed next.

Cao *et al.* (2019) theorise that shifting the decision boundary according to the class distribution may improve generalisation on minority classes. They

propose a label distribution-aware margin loss function to shift the decision boundary between clusters of different sizes. To ensure the decision boundary is appropriately adjusted, their loss function incorporates a theoretical margin-based generalisation bound. Furthermore, they propose a deferred data sampling method that allows the network to first learn feature representations in the standard way before training is adjusted to account for the class distribution by incorporating a modified sampling strategy. Their experiments show that these methods are effective on large scale and severely imbalanced data, with further improvement when both methods are combined.

Our work will investigate a simplistic regularisation technique as an alternative to constructing customised loss functions. Furthermore, the effects of different sampling strategies on network learning will be investigated, and an extension of the deferred sampling method will be considered.

Yang and Xu (2020) consider the value of the label information in class imbalanced data. They theorise that although labelled data is necessary in supervised learning settings, it is the label distribution which biases the network towards majority classes. As such, they investigate methods to either leverage the positive or circumvent the negative effects of label information. More specifically, semi-supervised methods such as pseudo labelling are proposed to improve the label distribution. By incorporating large amounts of external unlabelled data, the network can benefit from more balanced labelled data. Additionally, self-supervised learning is considered as a pre-training procedure to optimise the network on label-agnostic data. In the initial stage, labelled data is discarded and the network is tasked with learning some pretext task, whereafter the label information is reintroduced and standard supervised learning may commence. Both of their self- and semi-supervised methods demonstrate improved results when utilised for class imbalanced learning.

Incorporating additional labelled data with semi-supervised methods seems to be an appropriate approach for the class imbalanced problem. However, after extensive experimentation, Yang and Xu (2020) determine that semi-supervised methods are dependent on certain characteristics of the unlabelled dataset, and show that the distribution, size and data domain of the data are of importance. As an extension, we will investigate the effects of network confidence during the pseudo labelling process, when subjected to class imbalanced data, as well as realistic settings where both the labelled and unlabelled data may exhibit a long-tailed distribution. Network confidence is an important metric used in various machine learning applications such as semi-supervised learning tasks.

1.5 Objectives and thesis outline

This thesis aims to understand how neural network training will shift the decision boundary according to the distribution of the data, how network confidence is affected by class imbalance, whether there exist general solutions to adjust the decision boundary, and how large unannotated datasets may potentially help to alleviate the class imbalance problem.

Chapter 2 proceeds to define the datasets used for the following investigations, and state the experimental setup used throughout this thesis. Chapter 3 further investigates how the decision boundary of neural networks are influenced when learning on class imbalanced datasets. Chapter 4 determines the effects of class imbalanced learning on network calibration. In Chapter 5, regularisation methods are investigated as a means firstly to increase the number of diverse representations seen during training by generating synthetic samples, and secondly to indirectly shift the decision boundary according to the class distribution. In Chapter 6, data sampling techniques are considered for adjusting the decision boundary and improving generalisation of minority classes. In Chapter 7, semi-supervised methods are investigated as an alternative approach to traditional data collection, and as a possible solution for the class imbalance problem. In Chapter 8, a discussion is given of the overall findings of this thesis as well as prospective avenues for future work.

Chapter 2

Datasets and experimental setup

For this thesis we wish to make conclusions applicable to general image classification tasks, based on reproducible results. Various insights obtained in this work are gathered from empirical observations, and as such, steps are taken to encourage robust results. It should be noted that performing experiments on many different datasets with different levels of class imbalance for multiple network architectures are not feasible for our scope of work. Consequently, we limit our experiments to manageable datasets and networks often used for benchmarking in the literature.

2.1 Datasets

To reduce the cost of experimentation, we utilise smaller datasets that are artificially class imbalanced. The considered datasets are commonly used for research applications and are originally class balanced.

To simulate a class imbalanced scenario, datasets are artificially altered to follow an exponential distribution. The use of the exponential distribution follows common practice among current research in learning from class imbalanced data (Cui *et al.*, 2019*b*; Kang *et al.*, 2020; Zhou *et al.*, 2020). Although the exponential distribution does not have the same severe long-tailed imbalance as that of iNat2017, the performance of trained networks on minority classes should serve as a proxy. To construct the imbalance, samples are pseudo-randomly (determined by a seed value) selected from a dataset. We define the imbalance factor of a dataset as the number of training samples in the largest class divided by that of the smallest class, and denote it as the ρ value.

The CIFAR datasets offer a relatively straightforward and robust image dataset benchmark widely used in computer vision (Krizhevsky and Hinton, 2009). CIFAR is available in two variants, one consisting of 10 classes, known as

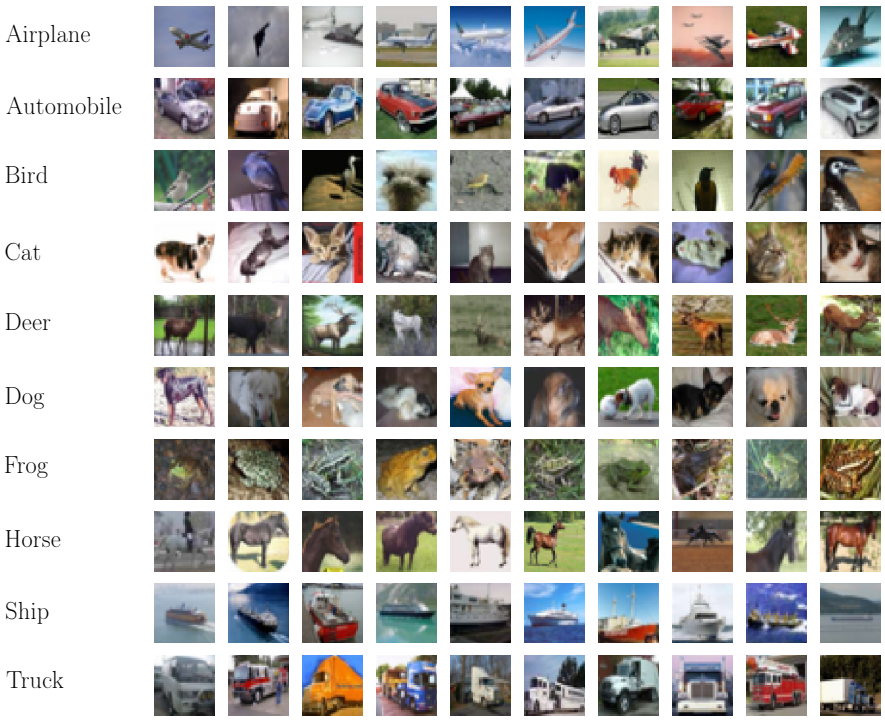


Figure 2.1: Ten random samples from each of the ten classes in the CIFAR-10 dataset.

CIFAR-10, and another containing 100 classes, known as CIFAR-100. Both sets have been manually selected from the 80 million tiny images dataset and independently labelled. The 80 million tiny images dataset is constructed from scaled down images extracted from the Internet (Torralba *et al.*, 2008). Both CIFAR variants are class balanced and consist of 60,000 32×32 colour images. The datasets are partitioned into 50,000 samples for the training dataset and 10,000 samples for the test dataset. Separate validation datasets are not included with these datasets.

The majority of experiments in this study utilise the CIFAR-10 dataset. To provide an indication of the general classification task considered, Figure 2.1 illustrates 10 random samples from each class. For the majority of samples only one object of interest is present in the centre of any particular image. The CIFAR-10 classes have been chosen to be mutually exclusive. As an example, the "truck" class only contains samples of large trucks, while the "automobile" class may include images of SUVs, and neither of these classes include samples of pickup trucks (Krizhevsky and Hinton, 2009). Consequently, the CIFAR-10 dataset may not present significantly complex or naturalistic images one would expect from real world observations, but the dataset does serve as an appropriate multi-class classification benchmark dataset for the scope of our work.

The class imbalanced version of CIFAR-10, with $\rho = 50$, will contain approximately 4,000 samples in the largest class and 84 samples in the smallest class, representing a relatively challenging scenario. For CIFAR-100, this reduces to 450 for the largest, and 9 for the smallest class. Not only is the minority class significantly more scarcely represented, but a vast majority of minority samples contain samples fewer than was in CIFAR-10. As a result, the imbalanced CIFAR-100 should prove to be a much more difficult classification task due to the large number of classes, and extremely scarce amount of data for the minority classes.

Another dataset, known as CINIC-10, extends the samples for the 10 classes of CIFAR-10 to 27,000 observations each (Darlow *et al.*, 2018). The additional images, drawn from the ImageNet dataset, are cropped and down-scaled to the image size of CIFAR-10, hence these images may not necessarily be as centred as the images of CIFAR-10.

The creators of CINIC-10 note that some of the added images may contain objects of interest that are less class-distinct than those of CIFAR-10. For example, images of goats and cows are included in the "deer" class. This can be seen as covariate shift, a form of distribution shift. Distribution shift occurs when the joint distributions of inputs and labels of a dataset differs from that of another dataset, while covariate shift describes the scenario where only the input changes. In realistic semi-supervised learning applications, it may be expected that incorporating additional data, such as CINIC-10, will exhibit some form of distribution shift from the original data, likely increasing the learning difficulty of the task at hand (Su *et al.*, 2021). The creators are unclear how severe this is, yet CINIC-10 should prove appropriate as an additional source for our investigation into semi-supervised methods in Chapter 7.

CINIC-10 is made available with train, validation and test partitions, each containing 90,000 samples. The CIFAR-10 train and test datasets are included within the corresponding partitions of CINIC-10. To ensure that the samples from the test dataset of CIFAR-10 are not accidentally mixed with the CINIC-10 training dataset, the CINIC-10 test partition is excluded in our experiments. As a result, the class size is reduced to 18,000 samples per class. This is still three times larger than the original CIFAR-10 dataset, and should be applicable for semi-supervised learning methods.

To simplify the scope of this work, we use class balanced validation and test datasets, and make the assumption that they are sufficient to describe the underlying distribution of considered classification tasks. This removes the need to incorporate additional measurements to account for class imbalanced test and validation datasets.

2.2 Experimental setup

The general experimental framework used throughout this thesis is outlined in the following section and these details will apply unless specified otherwise. Experimental implementations such as dataset pre-processing and model training are implemented within the TensorFlow environment (Abadi *et al.*, 2016).

The ResNet family of networks is widely used for benchmarking (He *et al.*, 2016). We incorporate a ResNet-34 variant, as it is often used for the datasets considered in this thesis. The network consists of 32 convolutional layers, partitioned into four differently sized convolutional blocks. To address the vanishing gradient problem typical of very deep neural networks (large numbers of stacked convolutional layers), ResNet includes skip connections between every second convolutional layer. The number of filters in convolutional layers increase from 64 to 512 in subsequent blocks, with convolutional filter sizes of 3×3 and ReLU as activation function. To aid model convergence, our particular version of ResNet-34 implements batch normalisation between convolutional blocks (Ioffe and Szegedy, 2015). The last convolutional block is followed by a global average pooling layer and a fully connected dense layer with softmax activation.

During training the input data is fed to the model in mini-batches of 64 samples before a network update is made, and in total, networks are trained for approximately 20,000 updates. The initial learning rate is set to 0.01 with a decay rate of 0.0001 applied after each network update. For the output of the classification layer, the softmax activation function is used, and the loss is computed as the categorical cross-entropy loss between the true and predicted labels. To improve convergence, a stochastic gradient descent optimiser is utilised with momentum set to 0.9.

Training may be sensitive to the initial parameter values, and to promote fair comparison, all networks are initialised with the same set of randomly generated weight and bias parameters. Furthermore, to reduce undesired variability, training sessions are repeated five times. To alleviate hardware constraints and increase throughput, the ResNet-34 parameters are stored and processed in 16-bit float precision.

When constructing an artificial dataset, the representation of the underlying class distribution may be altered (Sohn *et al.*, 2020), and the learning difficulty (granularity) between classes may be drastically changed (Cui *et al.*, 2019*a*). Hence, training on altered datasets may produce misleading results and influence conclusions. To address this, the class order as well as the sample order for each class in a dataset are pseudo-randomly shuffled before alterations such as distribution changes are made. To ensure consistency, each experiment is

performed on three differently shuffled datasets. The reported score for a particular experiment will thus be the averaged output from 15 training sessions, with the standard deviation included as well.

As minority samples may be especially valuable for model performance (Sohn *et al.*, 2020), we re-sample from the training dataset at random if the last batch of an epoch is not fully populated. To promote convergence, image pixel values are normalised to values between 0 and 1. For datasets that do not include a validation dataset, a pseudo-random selection of 15% is partitioned as a validation set from the training dataset. This is performed before a dataset is artificially imbalanced, hence all validation datasets approximate a class balanced distribution.

We are now in a position to investigate how learning from class imbalanced data may affect network training. In the next chapter, we investigate how the decision boundary may be influenced according to the distribution of the training data.

Chapter 3

The decision boundary

We proceed to investigate how the distribution of the training dataset may influence the decision boundary more closely, to better understand how decision boundaries are formed when learning from class imbalanced data. To start, consider a given classification task that optimises a network's parameters over the categorical cross-entropy loss between predicted and ground truth labels. This loss function is given as

$$L = -\sum_{n}\sum_{k} t_{k,n} \log\, p_k(x_n), \tag{3.1}$$

where $t_{k,n} \in \{0, 1\}$ is a ground truth label indicating whether the n^{th} image belongs to the k^{th} class. The term $p_k(x_n)$ is the softmax function that maps the outputs of the network to the range $[0, 1]$, representing the model's confidence that image x_n belongs to class k, as

$$p_k(x_n) = \frac{e^{(w_k^T \phi(x_n))}}{\sum_i e^{(w_i^T \phi(x_n))}}, \tag{3.2}$$

where w_k is the weight vector in the softmax layer for the k^{th} class, and $\phi(\cdot)$ denotes the network output for image x_n. For simplicity, bias parameters are included in weight vectors.

To better understand how the decision boundary behaves under imbalanced data, we consider the case of classifying a sample x as one of two classes. To determine the class allocation, the predicted probabilities between both classes are combined as

$$\frac{p_j(x)}{p_k(x)} = \frac{e^{(w_j^T \phi(x))}}{e^{(w_k^T \phi(x))}} = e^{[(w_j - w_k)^T \phi(x)]}, \tag{3.3}$$

where p_j and p_k denote the predicted probability of a sample x belonging to class j and k, respectively. It follows then that the decision boundary that separates these two classes will be a vector perpendicular to $w_j - w_k$.

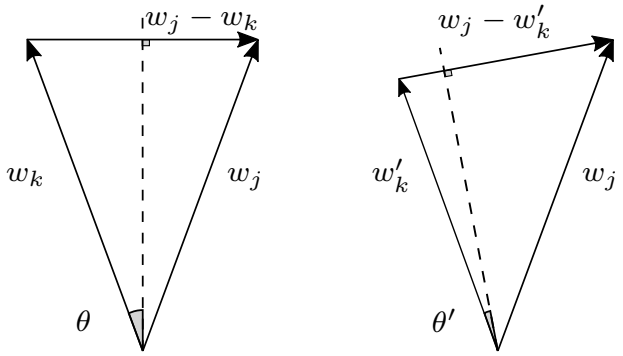


Figure 3.1: Relationship between the norm of weight vector w_k and the volume size of the partition for the k^{th} class. The dashed line represents the hyperplane (perpendicular to $w_j - w_k$) which separates the two adjacent classes. As shown, when the norm of w_k decreases ($||w'_k||_2 < ||w_j||_2$), the k^{th} class tends to possess a smaller volume size in the feature space ($\theta' < \theta$).

Figure 3.1 shows two scenarios that will produce different decision regions for these classes. The illustration demonstrates the connection between the norm of a class weight vector and the volume size of its corresponding region in feature space. For equally sized weight norms ($||w_k||_2 = ||w_j||_2$), the decision boundary divides the classification volume equally. However, as illustrated on the right of the figure, when the weight norm of class k is comparatively smaller ($||w'_k||_2 < ||w_j||_2$), the classification volume is decreased ($\theta' < \theta$), resulting in a smaller decision volume to predict class k.

Furthermore, for learning from imbalanced data, it can be demonstrated empirically that the class weight norms of a trained network's classification layer are correlated with class frequency. We compare a ResNet-34 network trained on two variations of CIFAR-100 and the weight norms for both in Figure 3.2. For a given classification layer, the L_2 norm of the weights for each class is calculated and sorted according to the class frequency in decreasing order. The first variation of CIFAR-100 is created by imbalancing the dataset with $\rho = 50$. The second is a balanced variation that has an equal number of samples per class removed from the original dataset, until it contains the same total number of samples as the first (imbalanced) variation. The choice of CIFAR-100 over CIFAR-10 here is to more clearly visualise the correlation over a larger number of classes.

Figure 3.2 shows that class frequency influences the weight norms in the classification layer of a network. The networks are obtained by re-training the classification layer of ResNet-34 that has been pre-trained on ImageNet. Keeping the model parameters of the hidden layers frozen allows us to produce the same

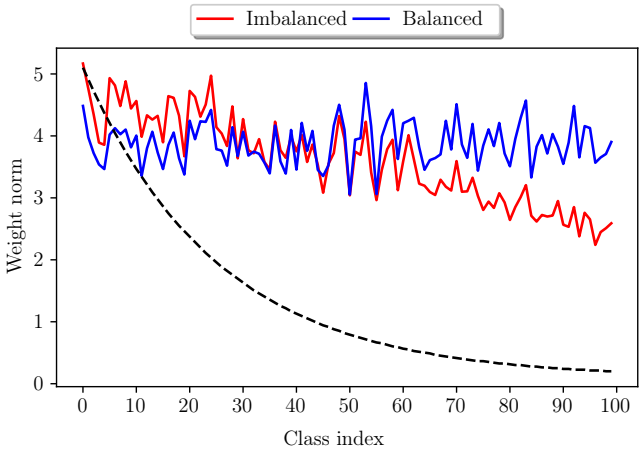


Figure 3.2: Comparison between the per-class classifier weight norms of a neural network trained on an imbalanced (red) and a balanced (blue) CIFAR-100 dataset. The networks are obtained by re-training the classifier of ResNet-34 that has been pre-trained on ImageNet. Classes are arranged according to the imbalanced class distribution in descending order, with the imbalanced distribution super-imposed as a dashed line for reference. As seen, classifier weight norms correlate with class frequency.

latent features when training on balanced and imbalanced data, and update only the classifier weights and biases. For training on a class balanced dataset, the class weight norms tend to be roughly uniform. However, training from class imbalanced data will tend to produce weight norms shifted in relation with the class imbalance, where the norms of majority classes are larger than those of the minority classes.

These results suggest that the decision boundary is correlated with the class distribution of the training data. The classification regions of the minority classes will tend to be smaller than those of majority classes, because of the disproportionate number of training samples. Using a network pre-trained on ImageNet may not be representative of a typical weight norm distribution, as the features have been effectively learned on a balanced dataset. Nevertheless, the aim of this demonstration is to show that the weights in the classification layer may change according to class frequency.

To explain why this correlation exists, Tang *et al.* (2020) state that the momentum component typically incorporated with stochastic gradient descent methods influences parameter updates to favour the more frequent classes. To clarify, we generalise the learning process to

$$v_t = \mu \cdot v_{t-1} + g_t, \quad \theta_t = \theta_{t-1} - l \cdot v_t, \tag{3.4}$$

where the variables in the t^{th} iteration are: model parameters θ_t, gradient g_t, velocity v_t, momentum decay ratio μ, and learning rate l.

Over the course of training, the momentum $\mu \cdot v_{t-1}$ typically stabilises gradients, dampening the oscillations caused by individual samples or mini-batches. When learning from a balanced dataset, the momentum will be equally contributed to by every class, however when learning from imbalanced data, the momentum will be dominated by samples from the majority classes. Furthermore, as the majority of samples will come from only a few classes, which might have relatively smaller variance in comparison to the under-represented minority classes, the momentum will guide the optimisation procedure to a local minimum that favours the majority classes. Tang *et al.* (2020) proceed to theorise that this causes a spurious correlation, and although their work is insightful, their proposed method is grounded in causal inference methods which are outside the scope of this work.

For applications reliant of accurate recognition of minority classes, it is therefore desirable to mitigate the effects of the class imbalance problem as much as possible. In the next chapter we investigate how network calibration is affected when learning from class imbalanced data.

Chapter 4

Confidence calibration

Apart from classifying with sufficient accuracy, classification models may also need to report on how confident they are in their predictions. The ability to indicate how confident a decision maker is, be it human or machine, can be paramount in applications where incorrectly interpreting a decision or misjudging the confidence behind a decision could cause harm.

As machine learning models integrate into real world decision making pipelines, the interpretability of their decisions in high-risk applications such as autonomous vehicle control (Levinson *et al.*, 2011) or automated healthcare (Miotto *et al.*, 2016) has become essential, with reliable confidence measures being no exception. Moreover, as deep neural networks have established supremacy in many pattern recognition tasks, it is the estimation of uncertainty in these types of classifiers that will be of increasing importance.

In recent years it seems deep neural networks have become overconfident (miscalibrated) in their predictions, meaning their predicted scores can be much higher than their accuracy (Guo *et al.*, 2017). More specifically, the probability of a sample belonging to a certain class by the softmax function will on average be higher than the likelihood that the sample actually belongs to said class.

To illustrate, the top row of Figure 4.1 provides confidence histograms for ResNet-34 models trained on different datasets. Confidence histograms plot the distribution of a model's predicted probabilities (prediction confidence) on the test set as a histogram separated into 10 equally sized bins (Dedduwakumara and Prendergast, 2020). Here the predicted probability is the score for the class the network deemed most likely to be correct. The left column of Figure 4.1 is produced by a model trained on a balanced CIFAR-10 dataset, while the right column indicates the results for a model trained on an imbalanced CIFAR-10 dataset with $\rho = 50$. Recall that ρ indicates the extent of imbalance for a dataset, determined as the number of samples in the largest class divided by the number of samples in the smallest class. Results are averages

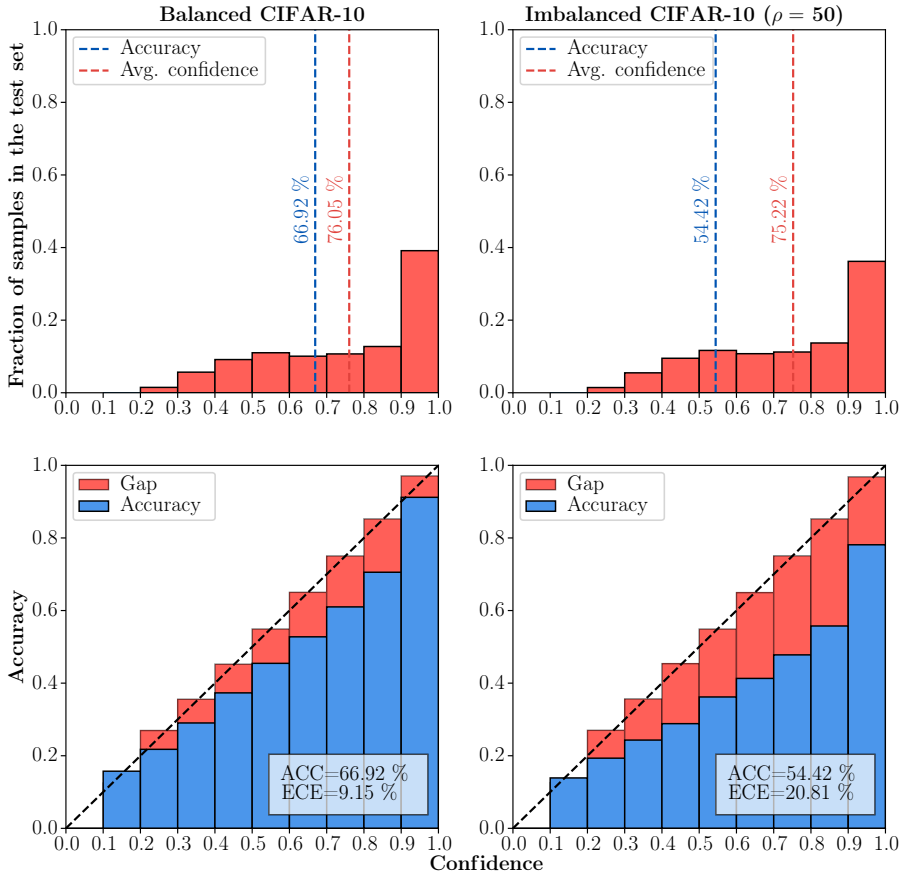


Figure 4.1: Confidence histograms (top) and reliability diagrams (bottom) for a ResNet-34 model trained on a balanced (left) and imbalanced (right) subset of CIFAR-10. Total samples are kept the same. Imbalanced learning leads to greater miscalibration.

attained from training over five runs on three folds of CIFAR-10, and all folds are constructed with the same total number of samples for fair comparison.

As the skewed histograms show, the trained networks tend to be highly confident in their predictions. Furthermore, by superimposing the test accuracy as well as the averaged confidence onto the histograms, the overconfidence is clearly visible.

Confidence histograms provide a simple visualisation of model confidence, but to better understand and study the role of confidence, a more precise defini-

tion for model calibration is required. Formally, we define a labelled dataset D to contain samples $X \in \mathcal{X}$ and labels $Y \in \mathcal{Y}$ as random variables that follow an underlying joint probability distribution $\psi(X, Y) = \psi(Y|X)\psi(X)$, and $\mathcal{Y} = \{1, \ldots, C\}$ where C is the number of classes. We let f be some neural network with output defined as $f(X) = (\hat{Y}, \hat{P})$, where $\hat{Y}$ is a class prediction made by the network, and $\hat{P} \in [0, 1]$ is the corresponding confidence from the network output by the softmax function.

Simply stated, we would like the confidence estimates $\hat{P}$ for all samples to be well-calibrated, meaning the network produces $\hat{P}$ as a true probability. For example, given 100 predictions from a perfectly calibrated model, each with a confidence of 0.95, we expect 95 of the predictions to be correctly classified. The notion of perfect calibration can then be written as

$$\mathbb{P}(\hat{Y} = Y|\hat{P} = p) = p, \quad \text{for all } p \in [0, 1], \tag{4.1}$$

where the probability is over the joint distribution.

The probability in (4.1) cannot be calculated with finite samples, and instead empirical approximations are needed to capture its essence. As a result, we define the expected calibration error (ECE) as a measurement for miscalibration (Naeini *et al.*, 2015). ECE operates on the notion that miscalibration is the difference in expectation between confidence and accuracy.

To approximate the expected accuracy from finite samples, sample predictions are partitioned into $M = 10$ equally sized intervals. Letting B_m be the set of indices of samples whose prediction confidence falls into the interval $(\frac{m-1}{M}, \frac{m}{M})$, the accuracy of B_m can be calculated as

$$\text{acc}(B_m) = \frac{1}{|B_m|} \sum_{i \in B_m} \mathbf{1}_{(\hat{y}_i = y_i)}, \tag{4.2}$$

where $\hat{y}_i$ and y_i are the predicted and true class labels for sample i, and $\mathbf{1}$ is an indicator function defined as

$$\mathbf{1}_{(\hat{y}_i = y_i)} = \begin{cases} 1 & \text{if } \hat{y}_i = y_i \\ 0 & \text{otherwise.} \end{cases} \tag{4.3}$$

The average confidence within B_m can be defined as

$$\text{conf}(B_m) = \frac{1}{|B_m|} \sum_{i \in B_m} \hat{p}_i, \tag{4.4}$$

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

Finally, the ECE of a model is calculated by taking a weighted average of the difference between accuracy and confidence of all B_m, giving

$$\text{ECE} = \sum_{m=1}^{M} \frac{|B_m|}{n} |\text{acc}(B_m) - \text{conf}(B_m)|, \tag{4.5}$$

where n is the total number of samples.

In an attempt to understand the role of miscalibration in modern neural networks, Guo *et al.* (2017) demonstrated the strong effects that widely adopted practices such as batch normalisation (Ioffe and Szegedy, 2015), increased network capacity as well as weight decay regularisation (Hanson and Pratt, 1988) can have on network calibration. They concluded that modern architectures are prone to miscalibration, and tend towards overconfident predictions.

This overconfidence is exacerbated when the training set is imbalanced. Recently, Zhong *et al.* (2021) found that models trained on long-tailed distributions tend to be more miscalibrated and overconfident, compared to models trained on balanced datasets. This phenomenon is demonstrated in Figure 4.1 where the confidence histograms and corresponding reliability diagrams (described below) are shown for a ResNet model trained on a balanced dataset in the left column and one trained on an imbalanced dataset on the right. By keeping the total number of samples constant in each dataset, and varying the distribution over class labels, the increase in miscalibration as a repercussion of imbalanced learning is concretely demonstrated. The results shown are averages from multiple model runs and several dataset configurations.

Reliability diagrams (shown in the bottom row of Figure 4.1) provide more comprehensive visualisations of model calibration than confidence histograms. These diagrams visualise the accuracy as a function of model confidence (DeGroot and Fienberg, 1983), and can be seen as a representation of the ECE metric. Concretely, reliability diagrams are computed by partitioning the expected accuracy and predicted confidences for a given set of samples into M equally sized bins B_m, similar to the computation of $\text{acc}(B_m)$ and $\text{conf}(B_m)$ for ECE, and plotted as histograms. The accuracies $\text{acc}(B_m)$ are shown as blue bars in reliability diagrams, while the difference between $\text{acc}(B_m)$ and $\text{conf}(B_m)$ represents the calibration gap (or error) and is shown in red.

A perfectly calibrated model should produce $\text{acc}(B_m) = \text{conf}(B_m)$ for all bins $m \in \{1, \ldots, M\}$. Therefore, for a well-calibrated model the heights of accuracy bars in a reliability diagram should be near the diagonal line, whereas any deviations from the diagonal would indicate miscalibration. As an example, in Figure 4.1, balanced learning produces accuracy bars that are lower than the diagonal line, indicating overconfidence. In our reliability diagrams we also

show the test accuracy and ECE score in the bottom right corner. Note that reliability diagrams do not display the proportion of dataset samples in a given bin; this is indicated in the confidence bars of the confidence histograms.

The calibration results in Figure 4.1 illustrate the additional effects of imbalanced learning on model calibration. Relative to a balanced scenario, the confidence histograms show a decrease in overall accuracy, reaffirming the difficulties of imbalanced learning stated in Chapter 1. Moreover, the reliability diagrams suggest expected accuracy declining in all intervals. It is worth noting that the average confidence does not change significantly (91.03% vs 90.67%). Perhaps more severely imbalanced scenarios, or analysing the confidence across various splits, could elucidate the matter.

Regardless, the clear increase in ECE and decrease in overall accuracy exemplify the deteriorating performance and instilled overconfidence. Intuition could suggest the increased overconfidence to be caused from the over-emphasised majority classes, while the decrease in accuracy could be from the unrecognised minority.

As an additional observation of the effects of miscalibration, Figure 4.2 (left) plots the classification error (as a percentage) and negative log-likelihood loss (scaled to the interval (0,100) for visualisation), as training progresses on an imbalanced dataset. Notice that the loss decreases in the initial phase of learning, but eventually starts to increase after a certain point, indicating the typical signs of overfitting. However, while the validation loss increases, classification accuracy on the validation set continues to improve (the error decreases).

This surprising behaviour where overfitting on loss is beneficial to classification accuracy can be evidence for miscalibration. In the middle and right of Figure 4.2, the model's calibration is evaluated at the lowest validation error (middle) and at the lowest validation loss (right). The comparison shows that early stopping based on validation loss may produce a relatively well-calibrated model, while early stopping based on validation accuracy may produce a more miscalibrated model but with better accuracy. It should be noted that this characteristic of overfitting only on the loss may occur when training on a balanced dataset as well.

For imbalanced learning, it is unclear how miscalibration manifests across the different classes. In Figure 4.3 we demonstrate a striking calibration bias towards the majority classes, where model calibration is near perfect on the majority classes, while less represented classes produce progressively more severe levels of miscalibration.

The confidence histograms in Figure 4.3 show that the distribution of con-

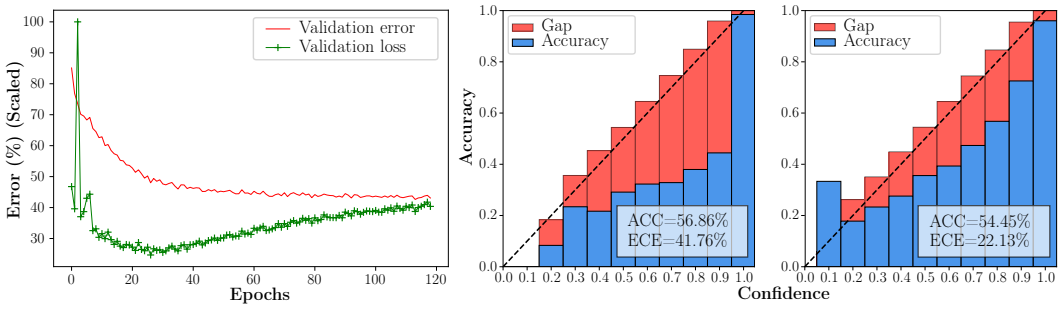


Figure 4.2: Left: validation error and (scaled) loss of a ResNet-34 model training on an imbalanced CIFAR-10 set. After 30 epochs the validation loss starts to increase while the error continues to decrease, indicating miscalibration. Middle and right: reliability diagrams for the network at the point of lowest validation error and lowest validation loss, respectively, suggesting calibration is traded for higher accuracy.

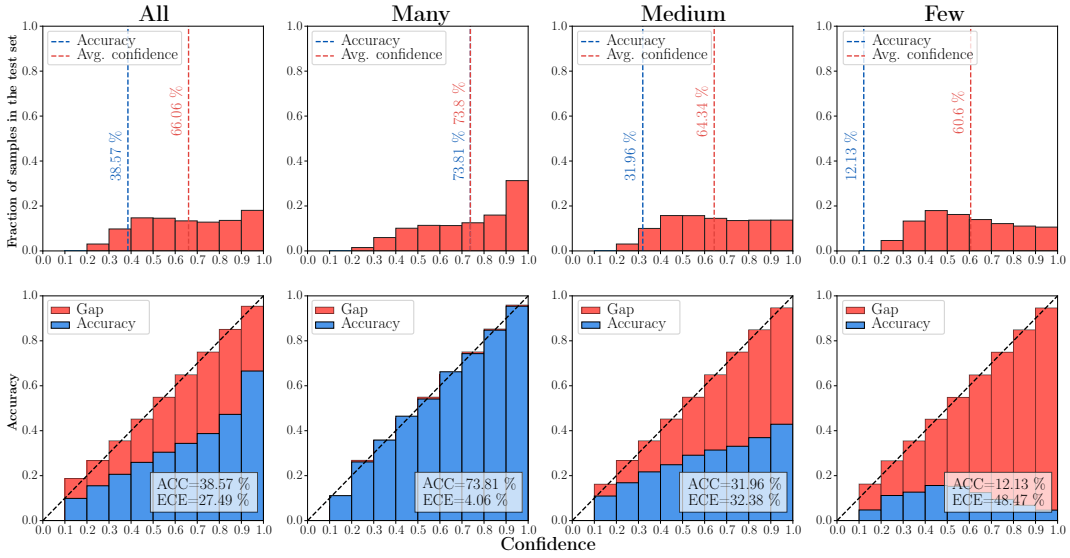


Figure 4.3: Confidence histograms (top) and reliability diagrams (bottom) for a model trained on imbalanced data, across different splits of the classes. The model achieves near perfect calibration on the majority classes, while under-represented classes lead to the highest overconfidence.

fidence scores depends on the distribution of class labels in the training set. Majority classes tend to receive higher confidence scores, while progressively lower class frequencies tend to yield lower confidence scores. This observation may exist as a result of the decision boundary biased towards minority feature space, hence majority classes will likely be further from the decision boundary than samples from the minority. Additionally, as the classifier's weight vector influences the magnitude of the softmax output, discussed in Chapter 3, larger weights may produce larger predicted probabilities.

Popular methods to re-calibrate classification networks have hinged predominantly around either rescaling the predicted probabilities after training on a hold-out set (Kumar *et al.*, 2019; Guo *et al.*, 2017; Niculescu-Mizil and Caruana, 2005), or introducing regularisation to act as a form of calibration penalty to the supervised learning objective (Zhong *et al.*, 2021; Thulasidasan *et al.*, 2019; Müller *et al.*, 2019).

Since the class imbalance problem may include scenarios where the validation set itself is imbalanced, we do not explore re-calibration on a hold-out set. Furthermore, Mozafari *et al.* (2018) find that some post-training methods do not work properly when the validation set is small in size, which could be an implication of imbalanced learning. Consequently, in the next chapter we consider regularisation methods as a means to improve model calibration.

Chapter 5

Regularisation techniques

Regularisation encompasses techniques that encourage generalisation. Models with poor generalisation can be said to have overfit the training data, meaning the model has learned and favours sample-specific features (characteristics) over their respective class characteristics.

Typically, regularisation techniques may be divided into either internal or external methods. Internal methods impose learning constraints on the internal structure of the model during training. Conversely, external methods enforce learning constraints prior to training and are typically applied on the data, meaning that the structure and domain of the data are taken into consideration.

The primary focus of this chapter is external regularisation. Common internal methods such as weight decay (Hanson and Pratt, 1988) and dropout (Srivastava *et al.*, 2014) provide computationally cheap regularisation solutions and are widely adopted for various tasks. However, including these methods in our experiments may muddle the analysis as they may take on confounding roles (Guo *et al.*, 2017; Zhong *et al.*, 2021). The only internal regularisation method we include is batch normalisation (Ioffe and Szegedy, 2015) as it is part of the ResNet architecture.

The techniques discussed in this chapter will be evaluated exclusively on improvements towards better calibration and generalisation. Regularisation techniques in general, including some of the proposed methods, could be effective for other tasks such as robustness training (Zheng *et al.*, 2016). For simplicity, however, these additional aspects will not be considered.

5.1 External regularisation

External regularisation techniques operate prior to or during training and is generally dependent on the data, meaning prior knowledge of the data domain is typically incorporated to improve generalisation. As the experiments in this work is focused on image classification, this definition may be reduced to forms of image data augmentation.

Image data augmentation has been shown to be effective regularisation methods. Some of the earliest and perhaps simplest forms of image data augmentation to show effectiveness for regularisation comes from simple image manipulations (Simard *et al.*, 2000; Shorten and Khoshgoftaar, 2019). Knowledge of the dataset domain enables manipulations to emphasise desired invariant details and create new semantic-preserving variations of the original data. These manipulations may consist of simple transformations such as flipping an image around an axis, translating the image, or rotating the image.

The effectiveness of image manipulations is sensitive to the dataset domain, and careless use could degrade the training process. For example, when classifying dogs, image rotations could produce varied examples of invariant features such as the relative position of a breed's eyes, but for digit recognition, large rotations of the number 6 could lead to samples too similar to the number 9.

Image manipulations thus need to take into consideration the magnitudes of distortion that should be applied. Strong magnitudes of manipulation may produce distorted samples that may hurt generalisation, while manipulations too weak may fail to sufficiently promote invariances within the data.

We refer to weak augmentations as a relatively safe transformation, meaning there is a high probability of preserving the label post-transformation. Strong augmentations refer to relatively unsafe transformations, meaning they have a high likelihood of not preserving the label post-transformation.

5.1.1 Weak augmentation

Certain basic image manipulations have proven useful on datasets such as CIFAR (Cao *et al.*, 2019). We follow Sohn *et al.* (2020) and apply some of these manipulations sequentially on the training dataset with relatively weak augmentation. The manipulations are chosen to alleviate positional biases that may be inherent in the data. As an example, dogs in the CIFAR-10 images could be predominantly centred, which may impose a localisation bias towards features in the centre of the image.

We apply two image transformations. The first is random horizontal flip-

ping, which simply flips images around the vertical axis with a probability of 50%. The second transformation is random horizontal and vertical translation, which shifts images in certain directions. Image translation is set to randomly translate images by up to a distance of 12.5% of their respective sizes. To ensure image sizes remain unchanged, images are extended by reflecting pixel values about the image edges.

5.1.2 Mixing data

Another form of data augmentation is combining existing samples within the dataset with one another. Procedures for mixing data vary, with some methods combining pixel-level information (Summers and Dinneen, 2019; Yun *et al.*, 2019), while others mix feature-level information in the hidden layers (Verma *et al.*, 2019).

Overall, mixing methods aim to produce new synthetic samples that encourage more difficult or diverse learning in a particular way. Several proposed mixing strategies have extended to mixing label information in conjunction with image samples. Although many mixing methods remain data-dependent, they are easily implemented, cost effective and can be used in combination with other regularisation techniques.

One such method is mixup, a data augmentation scheme that combines images convexly in conjunction with their respective label information (Zhang *et al.*, 2018). In essence, mixup generates new artificial data by linearly interpolating between two randomly chosen sample-label pairs.

Mixup is principally a pre-processing procedure, where all images are transformed within a given batch before being fed to the training process. This process may however be implemented in an online fashion, to reduce memory requirements.

When mixing samples, the magnitude of the interpolation is determined by λ, a scalar value ranging between $[0, 1]$. To determine the value of λ, Zhang *et al.* (2018) propose to sample from some beta distribution (Sinharay, 2010).

Let B denote a batch of data containing m sample-label pairs X and Y, where the labels in Y are one-hot encoded. For two randomly chosen sample-label pairs (x_i, y_i) and (x_j, y_j) drawn from B, artificial data (x^r, y^r) is generated as

$$x^r = \lambda x_i + (1 - \lambda)x_j, \tag{5.1}$$

$$y^r = \lambda y_i + (1 - \lambda)y_j. \tag{5.2}$$

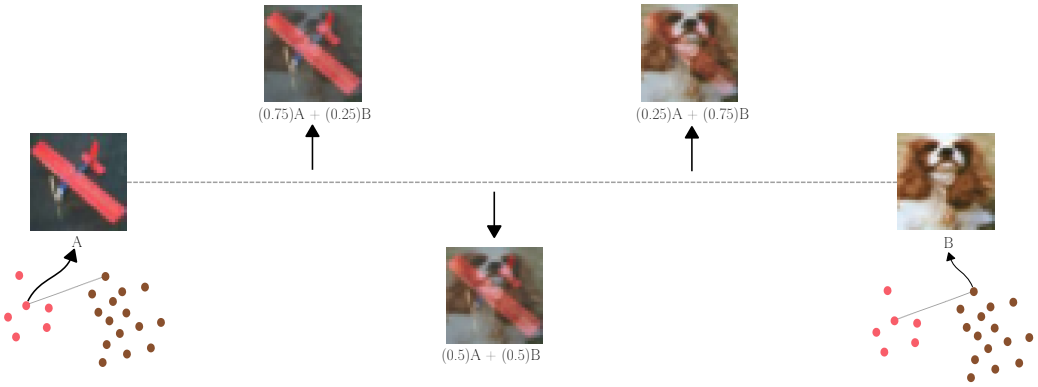


Figure 5.1: An illustration of mixup interpolations. Synthetic image-label pairs are convexly generated via linear interpolation between two randomly chosen samples. The point of linear interpolation is determined by λ, sampled from some beta distribution. Note that two samples can be chosen from the same class.

Figure 5.1 demonstrates various samples generated along the linear line of interpolation for two image-label pairs. For a classification dataset containing classes "airplane" and "dog", mixup selects two samples labelled as y_i and y_j. To determine the strength of interpolation λ is sampled from the beta distribution, where the variables α and β are pre-determined, typically via hyperparameter tuning. For this illustration, we omit the distribution and simply illustrate interpolations for a few values of λ between 0 and 1.

Intuitively λ may serve as a proxy for augmentation safety. This holds regardless of the label mixture, meaning for samples mixed from the same class (within-class), or different classes (between-class), convex combinations of natural images do not produce natural images for λ values close to 0.5.

To control the safety of augmentation, determining an appropriate λ is thus necessary. Zhang *et al.* (2018) choose to sample from a symmetric beta distribution, written as

$$\lambda \sim \frac{x^{\alpha-1}(1-x)^{\alpha-1}}{\int_0^1 t^{\alpha-1}(1-t)^{\alpha-1}dt}\,, \tag{5.3}$$

where $\alpha \in [0, \infty)$ and $x \in [0, 1]$.

They constrain their hyperparameter search space for α values between $[0, 1]$, where $\alpha = 1$ reduces the distribution to a uniform one. The authors note that although larger values for α are possible, increasing the likelihood of severe mixing may not provide safe augmentation of the original classes for learning and thus degrades performance.

Figure 5.2 compares the decision regions (indicated as shading) of a simple

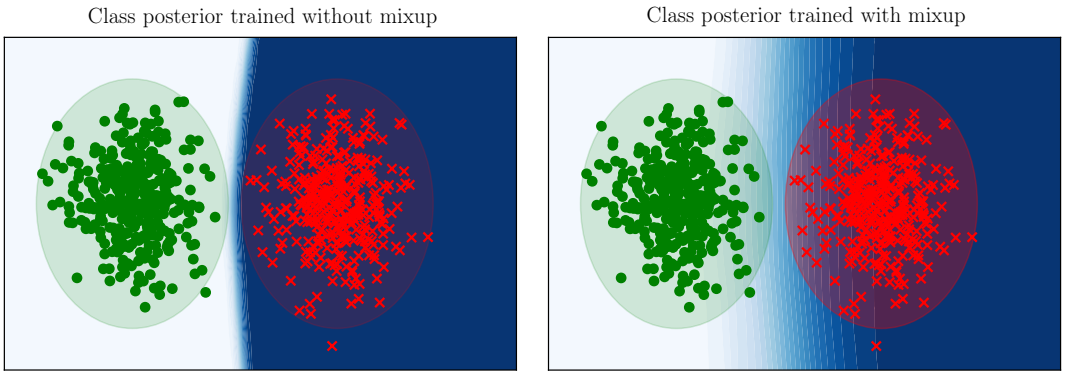


Figure 5.2: Effect of mixup with $\alpha = 1$ on a balanced toy problem, with class 0 shown in green and class 1 in red. The shades of blue indicate $p(y = 1|x)$. Left: without mixup, training produces a clear decision boundary. Right: with mixup, training leads to smoothed decision boundary regions between different classes.

neural network trained with and without mixup. This visualisation depicts the output probability of a sample belonging to the positive class, $p(y = 1|x)$. The toy example demonstrates the benefit of mixup training: training with mixup produces a decision boundary that transitions more gradually between classes. This provides a smoother estimate of uncertainty that should improve calibration.

One of the key operations that provides the smoothed boundary is that synthetic data is produced from selecting any samples. Mixing data from different classes may result in the creation of inputs that exist outside of the vicinity of the original class distributions, as indicated in Figure 5.1. For mixtures between points A and B, indicated by the line connecting the two points, many would populate outside of both empirical distributions.

Mixup may incur harmful augmentations as generated samples could unintentionally overlap with other class distributions, producing incorrectly labelled data. Additionally, the sampling strategy does not cater for imbalanced classes, and will inevitably be biased towards frequent classes.

For balanced datasets, research has been done into improved sampling methods (Guo *et al.*, 2019; Summers and Dinneen, 2019). We will investigate other sampling methods to account for imbalanced distributions specifically, namely by adjusting the label assignment method, explored next, or by adopting a more optimal batch sampling method, explored in Chapter 6.

5.1.3 Margin-based regularisation

Improving generalisation by promoting large decision boundary margins has been studied extensively and can be seen as a form of regularisation (Bartlett *et al.*, 2017; Cao *et al.*, 2019). We define the decision boundary margin as the shortest distance of any point in latent feature space to the decision boundary.

Remix is a label distribution-aware variant of mixup (Chou *et al.*, 2020) that follows the ideas of Cao *et al.* (2019) to modify class-specific margins in order to promote better generalisation of the minority classes. It follows the same sample mixing scheme as mixup, but introduces a distribution-aware label smoothing technique that promotes larger minority class margins, at the expense of majority class margins. Two hyperparameters are incorporated, to be used in conjunction with λ to determine if two chosen samples come from imbalanced classes, and whether the imbalance is severe enough to warrant intervention.

The formulation to determine a distribution-aware label λ_y, is as follows:

$$\lambda_y = \begin{cases} 0, & \text{if } \frac{n_i}{n_j} \geq \kappa \text{ and } \lambda < \tau, \\ 1, & \text{if } \frac{n_i}{n_j} \leq \frac{1}{\kappa} \text{ and } 1 - \lambda < \tau, \\ \lambda, & \text{otherwise,} \end{cases} \tag{5.4}$$

where $\lambda_y \in [0, 1]$, $\tau \in [0, 1]$ and $\kappa \leq 1$.

In the above, n_i is the total number of samples for class i. If the imbalance between two classes $\frac{n_i}{n_j}$ is greater than a predefined threshold κ, then the minority label should be considered, but the augmented sample will only be hard-labelled as the minority class if the sample mixture λ is below a predefined threshold τ. If λ is above τ, then the label will be smoothed by λ instead and the method defaults back to standard mixup.

An illustration of how remix is implemented between imbalanced classes is given in Figure 5.3. The choice of τ thus serves as intervention for the remix labelling procedure, encouraging the threshold to be moved in a certain direction given certain criteria are met. The frequency of when this intervention happens is of importance. If no imbalanced pairs (larger than κ) are presented or if the mixing factor is too small (lower than τ), then the relabelling procedure will have few opportunities to encourage the decision boundary shift.

Figure 5.4 compares the decision regions (indicated as shading) of a simple neural network trained without mixup or remix (left), and with remix (right) on an imbalanced toy dataset. This dataset is constructed by selectively removing samples from the toy dataset used in Figure 5.2 to simulate a scenario

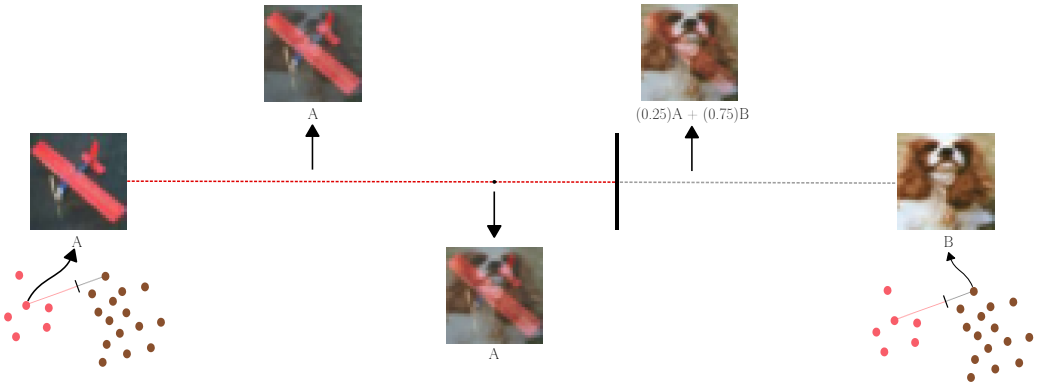


Figure 5.3: An illustration of how the hyperparameter τ affects remix labelling. Dots are majority and minority samples in the feature space. The dashed line represents all the possible mixed samples and the solid black line represents the threshold τ. When $\tau = 0$, mixed samples are labelled as in the original mixup. When $\tau > 0$, part of the mixed samples on the red dashed line will be labelled as the minority class. In the most extreme case against imbalance, τ is set to 1 where all mixed samples are labelled as the minority.

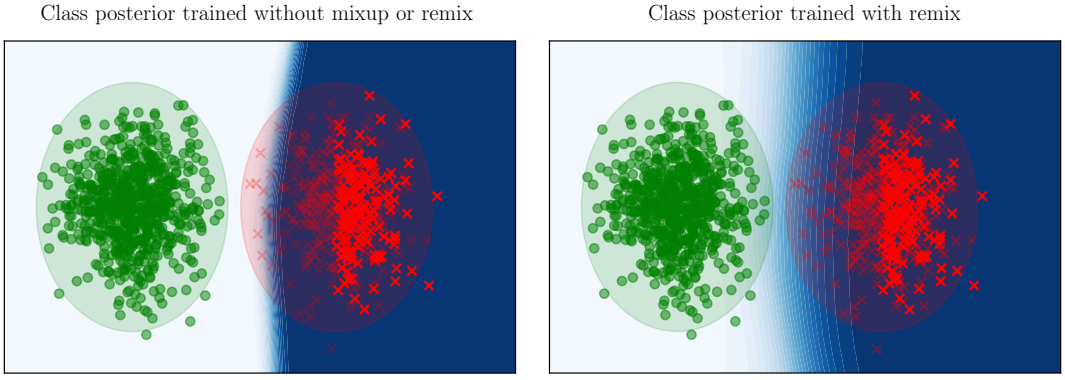


Figure 5.4: Demonstration of decision boundary shift due to learning from imbalanced data (left) and remix regularisation (right).

where the minority class does not contain enough samples to fully describe the underlying distribution. For illustrative purposes, the removed samples are shown in a darker red.

As can be seen, although the neural network without remix successfully classifies the data, the produced decision boundary encroaches on the underlying distribution of the minority class. Failure to account for the underlying distribution may stifle generalisation on the minority class, with new samples potentially misclassified as the majority. When remix regularisation is applied, the decision boundary is shifted to better account for the underlying minority class distribution. Following predefined hyperparameters, certain synthetic mixtures between the two classes are relabelled as the minority class to encourage larger minority class margins during training. In addition, remix training manages to produce a more gradual decision boundary inherent from mixup.

The application of remix is underpinned by an assumption that certain classes are under-represented, and to some undetermined degree, correlates with the relative difference in class sizes. With the hyperparameters influencing all relations between classes similarly, this may diminish certain decision boundaries.

In response, Chou *et al.* (2020) note that it may be beneficial to optimise a set of κ and τ individually for all classes, but optimising over such sets may be intractable for large multi-class classification datasets. Their findings conclude that for datasets such as CIFAR-10, general improvements can be obtained from simply optimising over all classes.

5.2 Comparison of regularisation techniques

In this section we compare the above-mentioned regularisation methods to better understand their effects on learning with imbalanced data, with specific focus on model accuracy and calibration. Our aim is to gain insights in order to further improve methods for learning with imbalanced data.

The experimental setup for model and dataset construction is described in Section 2.2. All model versions are trained for the same number of batch updates (20,000 updates), and evaluated on the epoch with the lowest validation loss. Results are obtained by averaging a total of 15 training sessions for each method. These sessions are produced by training on three randomly generated subsets of CIFAR-10, with $\rho = 50$, each trained five times. To simplify comparisons, we omit showing the performance of some methods, such as using only weak augmentation or remix, as their results are in line with the others.

To determine the optimal hyperparameter values for mixup and remix, a cost-

effective approach is utilised. The value of α for both is set to 1. The values for τ and κ are set to 0.5 and 3, respectively. We adopt these values as they have been tested on CIFAR-10 for both balanced and imbalanced cases and are recommended (Zhang *et al.*, 2018; Chou *et al.*, 2020).

Figure 5.5 presents the overall effects of regularisation for imbalanced learning with reliability diagrams and confidence histograms. The regularisation methods display consistent improvements in terms of generalisation and calibration compared to no regularisation (the baseline).

The most noticeable benefits are the supplementary improvements when combining regularisation methods. This is indicated by mixup-aug and remix-aug, the two best performing methods which have weak augmentation applied prior to sample mixing. Combining weak augmentation with sample mixing acts as a complementary regularisation combination. Image manipulations from weak augmentations will promote invariant features that are effective for feature learning, while sample mixing may maintain these emphasised features when mixing with other samples.

Furthermore, the results show that learning from invariant features produced by image manipulations tends to produce more confident predictions, which mixup alone does not achieve. This is indicated by the different predicted probability distributions (the confidence histograms in Figure 5.5) of mixup and mixup-aug, where the latter has a left-skewed distribution, while the former does not. It can be argued that left-skewed distributions such as these, produced by a well-calibrated model, would be desired as the model would then be both confident and correct about most of the predictions.

Although not as effective, applying mixup regularisation without weak augmentation does provide generalisation and calibration improvements. It seems that smoothing the decision boundary between classes does in fact act as a means to combat overconfidence, illustrated by the lower ECE score.

Mixup does not seem to provide a noticeable change in the distribution of predicted probabilities. This suggests that although mixing samples with mixup may be effective for improving calibration, the mixup augmentations may not provide sufficient samples to effectively learn invariant features from.

In our experimental setup, remix improves upon mixup regardless of whether weak augmentation is applied. The reliability diagrams show that remix training provides slightly higher overall accuracy and lower ECE than mixup. This indicates that increasing class margins may promote better generalisation and calibration. However, it is unclear from the reliability diagrams whether the improvements are indeed from indirectly increasing the minority class margins.

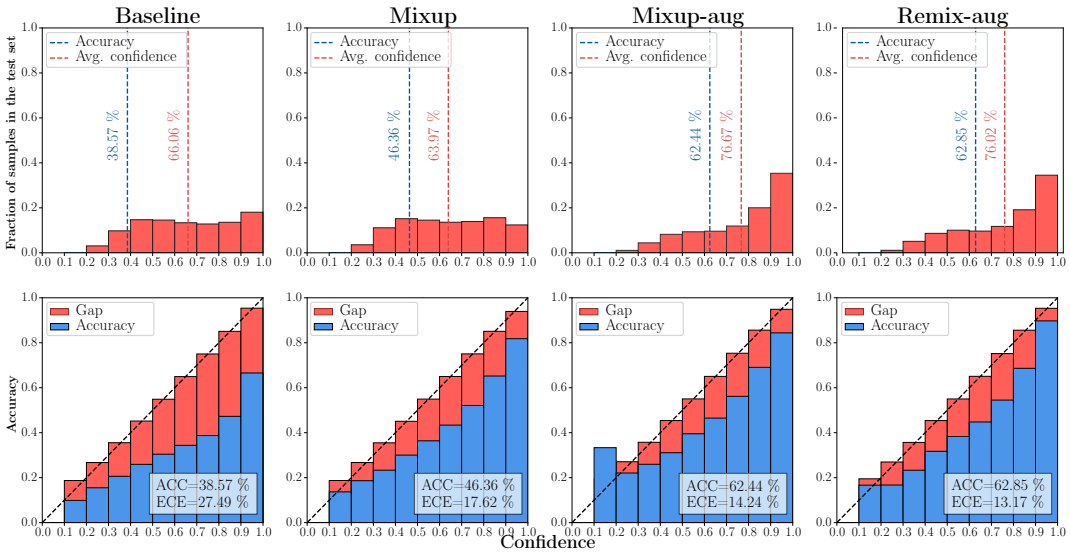


Figure 5.5: Performance comparison between different augmentation strategies. The baseline implements no image augmentation. All methods improve both calibration and generalisation. Combining mixup or remix with weak augmentation delivers higher gains. Remix with weak augmentation performs slightly better than mixup with weak augmentation.

To further compare the impact of regularisation on learning from imbalanced data, the test accuracy performance between different class splits is investigated. Table 5.1 reports the test accuracy of the proposed methods across various splits, as described in Section 1.3.

These results confirm that the discussed regularisation methods manage to improve generalisation, with accuracy gains over the baseline seen across all class splits. The baseline is produced by not applying any image augmentation strategies. The performance improvements among all of the splits substantiate the notion that neural networks are prone to overfitting, regardless of class frequency. Additionally, among the different splits, it seems that the largest improvement may be on the minority classes.

Comparing remix with mixup shows that remix improves the performance on the minority classes. For this particular dataset, sample pairs from classes with imbalanced ratios larger than κ may have been selected from the few as well as the medium splits. As a result, for classes in both of these splits there is an increase in accuracy over the mixup results, implying that for some of the classes, the respective margins have increased for better generalisation.

Remix seems to achieve increased accuracy on minority classes at the cost of

Method	All	Many	Medium	Few
Baseline	38.6 ± 1.0	73.8 ± 0.9	31.9 ± 3.4	12.1 ± 4.2
Weak augmentation	60.7 ± 0.1	87.5 ± 0.1	58.5 ± 1.8	37.0 ± 2.9
Mixup w/o aug	46.4 ± 0.7	77.9 ± 0.7	41.7 ± 1.7	21.0 ± 1.7
Mixup-aug	62.4 ± 0.3	**87.6 ± 0.1**	62.7 ± 2.4	36.9 ± 0.5
Remix-aug	**62.9 ± 0.4**	86.8 ± 0.2	**63.3 ± 2.8**	**38.2 ± 6.9**

Table 5.1: Split accuracy performance among different augmentation strategies. Results are shown as the averaged accuracy with the standard deviation for a particular split. The baseline implements no image augmentations. Highest score per split is in bold. Combining weak augmentation with remix achieves highest overall performance.

accuracy on the majority classes. This demonstrates that if margins for the minority classes are increased too aggressively, the resultant decision boundary may inadvertently hamper generalisation on more frequent classes. Therefore care should be taken to optimise λ, τ and κ appropriately.

Although not shown here, the original observation of overfitting exclusively on the validation loss (demonstrated in Figure 4.2) seems to be successfully mitigated when mixing methods are applied. Strong regularisation methods such as mixup have been shown to be effective towards combating overfitting (Carratino *et al.*, 2020), an aspect further investigated in Chapter 6.

In the presented setup, mixup and remix successfully mitigate significant signs of overfitting, increasing the number of additional batch updates before the training and validation losses start to diverge. We do not increase the training steps for the experiments here, which could lead to even better performance, as we wish to fairly compare the different aspects these methods have on an imbalanced dataset.

Another extension from this work is the impact on calibration when increasing minority class margins indirectly through remix. Intuitively, mixup achieves better calibration by smoothing one-hot labels, while the τ hyperparameter used in remix revokes the process, encouraging the model to be more confident towards certain classes.

Figure 5.6 compares the calibration metrics across splits between mixup and remix. For our dataset and settings, remix improves the calibration on classes in the few split, while deteriorating the calibration of classes in the many split. This result could stem from the improved generalisation gained when increasing class margins, that outweighs possible overconfident predictions. This is justified by examining the confidence histograms between class splits. For the few class split there is a slight increase in the averaged confidence, indicat-

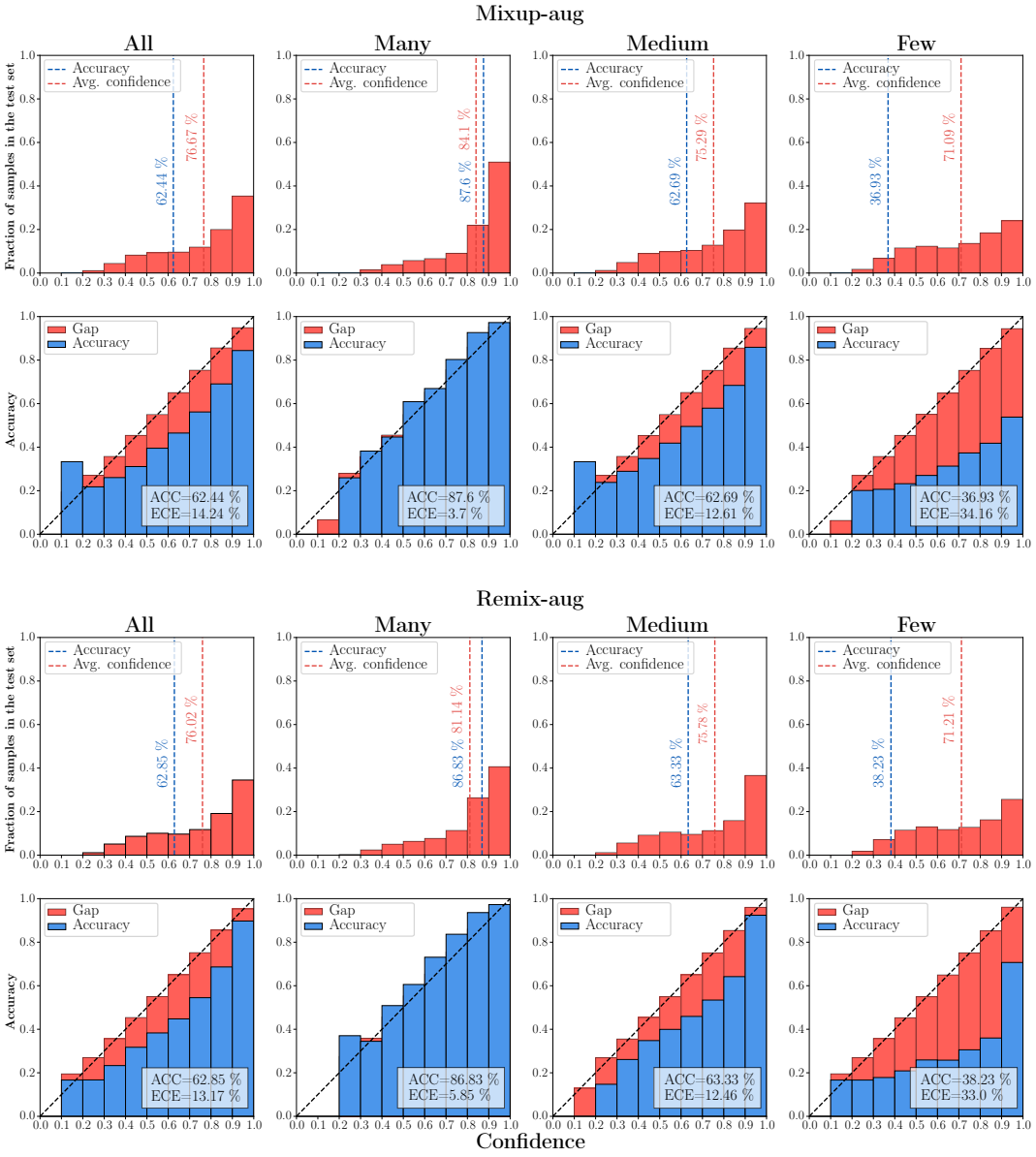


Figure 5.6: Comparison of calibration metrics for different class splits between mixup-aug (top) and remix-aug (bottom) methods. The remix-aug method reduces the ECE score on classes in the few split, while increasing the ECE score on classes in the many split. The compromise of calibration on majority classes are outweighed by the improvement on the minority classes. As a result, the remix-aug method improves network calibration over mixup-aug.

ing that the label smoothing regularisation is eased for the few class samples. However, the average accuracy is improved, indicating that the margins have increased, and as a result, the ECE score for the split is improved when remix is applied. The observed increase in confidence is almost negligible, however, and could be produced from other stochastic factors in the learning process.

Conversely, for the many class split, the network trained with remix produces a smaller averaged confidence score, and lower accuracy as compared with mixup, which produces a higher ECE score. The observed changes in confidence are marginal and could by influenced by the stochastic nature of the learning process. Overall, however, the compromise of calibration on majority classes is outweighed by the improvement on the minority classes.

In conclusion, the regularisation methods presented in this chapter manage to improve model calibration and generalisation. Combining weak augmentation with a data mixing method provides the greatest benefit, hence we incorporate weak augmentation as a prior regularisation method for all further experiments, and will reference the data mixing methods when needed. Mixup manages to improve generalisation, but the most prominent improvement is in model calibration. Through additional hyperparameter tuning, remix can offer additional improvements on both generalisation and calibration compared to mixup. The improvements are not as substantial as they could be, calling for additional techniques.

In the next chapter we investigate methods to address the class imbalance problem more directly, with an emphasis to further improve upon a possible biased decision boundary. In addition to these methods, the regularisation techniques of this chapter will prove to be complementary and easily incorporated in the learning scheme, forming part of the final approach to produce more effective models.

Chapter 6

Algorithmic methods

For machine learning methods operating on imbalanced data, numerous techniques have been studied. In general, these techniques aim to reduce the tendency of network predictions to bias towards majority classes and increase generalisation performance on minority classes (Leevy *et al.*, 2018; Johnson and Khoshgoftaar, 2019). In Chapter 3 it was demonstrated that the classification layer of a neural network tends to follow the distribution of the training data, influencing the decision boundary. In this chapter we will investigate methods to improve generalisation on minority classes and network calibration by adjusting the classifier weight norms.

Techniques to address the class imbalance problem can be grouped into either internal or external methods. Popular internal methods such as cost-sensitive re-weighting will assign different costs, or weights, to the loss. The terms in the loss function can then be catered to match a certain distribution and promote better generalisation on minority classes (Cao *et al.*, 2019).

Classical re-weighting methods assign class-specific weights to loss terms according to the inverse of class frequency. These methods however demonstrate convergence issues on highly imbalanced datasets, as well as typically severe compromises on the majority class generalisation (Mikolov *et al.*, 2013; Kang *et al.*, 2020).

Instead of applying costs proportional to frequency, more intricate methods have been proposed to weigh according to specific samples for more fine-grained control of the loss (Cui *et al.*, 2019*b*; Lin *et al.*, 2020), with Cao *et al.* (2019) further designing a label distribution-aware margin (LDAM) loss to encourage larger margins for minority classes. More specifically, LDAM normalises network outputs of each class according to the distribution of the data and a theoretical margin-based generalisation bound.

6.1 Sampling methods

In contrast to internal intervention, popular external methods manipulate the sampling procedure of data during training. The most straightforward of these methods aims to ensure that the class frequency for a given batch of data follows some desired distribution. We investigate sampling methods more thoroughly and aim to incorporate an appropriate sampling strategy to address the class imbalance problem.

The simplest forms of sampling are known as random under-sampling, which randomly discards or ignores samples from certain classes, and random over-sampling, which randomly duplicates certain samples. Each of these methods has benefits and drawbacks.

Ignoring samples in random under-sampling may reduce the total amount of data from which a model learns, and decreases the computational cost of training. However, for sufficiently large and severely imbalanced datasets, this approach may discard valuable and informative samples from the dataset, producing inferior performance compared to standard training (Buda *et al.*, 2018). To remedy this, more sophisticated discarding methods have been recommended, with some selectively identifying and discarding the least representative samples from a given class, in order to promote more effective and efficient training (Leevy *et al.*, 2018).

6.1.1 Over-sampling

As opposed to under-sampling, over-sampling methods populate the dataset to a desired distribution, often with duplicated images. These methods are however prone to certain disadvantages that may hinder effective training. A well-known issue is that of overfitting, where on sufficiently large and severely imbalanced datasets, a trained model will likely overfit on the duplicated samples (Leevy *et al.*, 2018).

To combat overfitting, image augmentation techniques may be incorporated to act as a form of additional regularisation. The synthetic minority over-sampling technique (SMOTE) (Bowyer *et al.*, 2002) is a well-known over-sampling method that augments the image set with a nearest-neighbour heuristic. More specifically, SMOTE generates synthetic samples by randomly interpolating linearly between certain dataset samples and a random selection of the N nearest samples of the same class.

Bellinger *et al.* (2021) combine mixup with over-sampling, and report additional improvements in terms of calibration and generalisation. Chou *et al.* (2020) use over-sampling with remix, postulating that this will increase the

frequency of interventions for the remix method to increase the boundary of the minority classes. They report improved generalisation over standard over-sampling.

There are various over-sampling strategies to achieve different training distributions. We investigate two of them to achieve class parity, and utilise mixup and remix regularisation to mitigate possible overfitting. Let n_j denote the number of samples for class j, and $n = \sum_{j=1}^{C} n_j$ the total number of training samples. The considered sampling strategies can be defined by the probability p_j of sampling an image from class j, following

$$p_j = \frac{n_j^q}{\sum_{i=1}^{C} n_i^q} \ , \tag{6.1}$$

where $q \in [0, 1]$ and C is the number of classes.

When no over-sampling is applied, a mini-batch is created by uniformly sampling, with replacement, from all instances within the dataset. This default sampling scheme is referred to as instance-balanced sampling (IBS), and has been the default method in previous chapters. The IBS scheme can be written as

$$p_j^{\text{IBS}} = \frac{n_j}{\sum_{i=1}^{C} n_i}, \tag{6.2}$$

where $q = 1$.

The over-sampling method used for SMOTE and other discussed methods sample among classes. Class-balanced sampling (CBS) can be described as first sampling uniformly, with replacement, among classes, and afterwards sampling an instance from each of those classes. The CBS scheme can be written as,

$$p_j^{\text{CBS}} = \frac{1}{C}, \tag{6.3}$$

where $q = 0$.

Progressively-balanced sampling (PBS) is a combination of IBS and CBS, and transitions from one to the other as training progresses. This mixed sampling approach is, as a result, dependent on the total number of epochs T during training, and is defined as a function of the current epoch t as

$$p_j^{\text{PBS}}(t) = \left(1 - \frac{t}{T}\right) p_j^{\text{IBS}} + \frac{t}{T} p_j^{\text{CBS}} \ . \tag{6.4}$$

Apart from overfitting on minority classes, sampling methods have been shown

to produce networks that underfit the training distribution as well, posing as an additional barrier for effective learning from imbalanced data. This is discussed in the next section.

6.1.2 Decoupled sampling

An implication of adjusting the classifier weight norms through sampling and re-weighting methods is that it may impair the representation learning process. It has recently been shown that altering the training distribution may actually produce varying qualities of learned features, and popular CBS methods learn sub-optimal representations (Zhou *et al.*, 2020; Kang *et al.*, 2020).

To demonstrate, we follow Kang *et al.* (2020) in order to compare the quality of learned representations and classifiers subject to different sampling strategies. From the findings, we wish to select the most appropriate method as our decoupled sampling procedure.

Concretely, we separately train the classifier (fully connected layers) and feature extractor (convolutional base) of a ResNet-34 network on a class imbalanced dataset. As we seek to find the most effective method, we implement remix regularisation with weak augmentation explained in Chapter 5. Results are reported as the averaged accuracy over five iterations of three random subsets of the CIFAR-10 dataset with $\rho = 50$. For all experiments, the network is reinstated with the parameters that produce the lowest validation loss. The rest of the dataset construction and network implementation follows the experimental setup as described in Section 2.2, unless stated otherwise.

In the first stage of training, different feature extractors are obtained by training the network under various over-sampling methods. These methods are standard instance-balanced sampling (IBS), class-balanced sampling (CBS), and progressively-balanced sampling (PBS). Networks are trained with standard mini-batch gradient descent for 20,000 batch updates, with a learning rate of 0.005.

In the second stage of training, the feature extractor is kept fixed while the classifier is re-trained under a specific sampling scheme. More precisely, the classifiers are re-initialised to the same random weights and bias parameters. The parameters of the feature extractor are then kept fixed (frozen). The classifiers are re-trained under different sampling strategies, effectively forcing the network to classify the presented features. For this stage, the learning rate is increased to 0.05, and networks are trained for 6,500 batch updates. The choice of batch updates follows the recommendation of Kang *et al.* (2020).

To summarise, the decoupled sampling method will separate the network train-

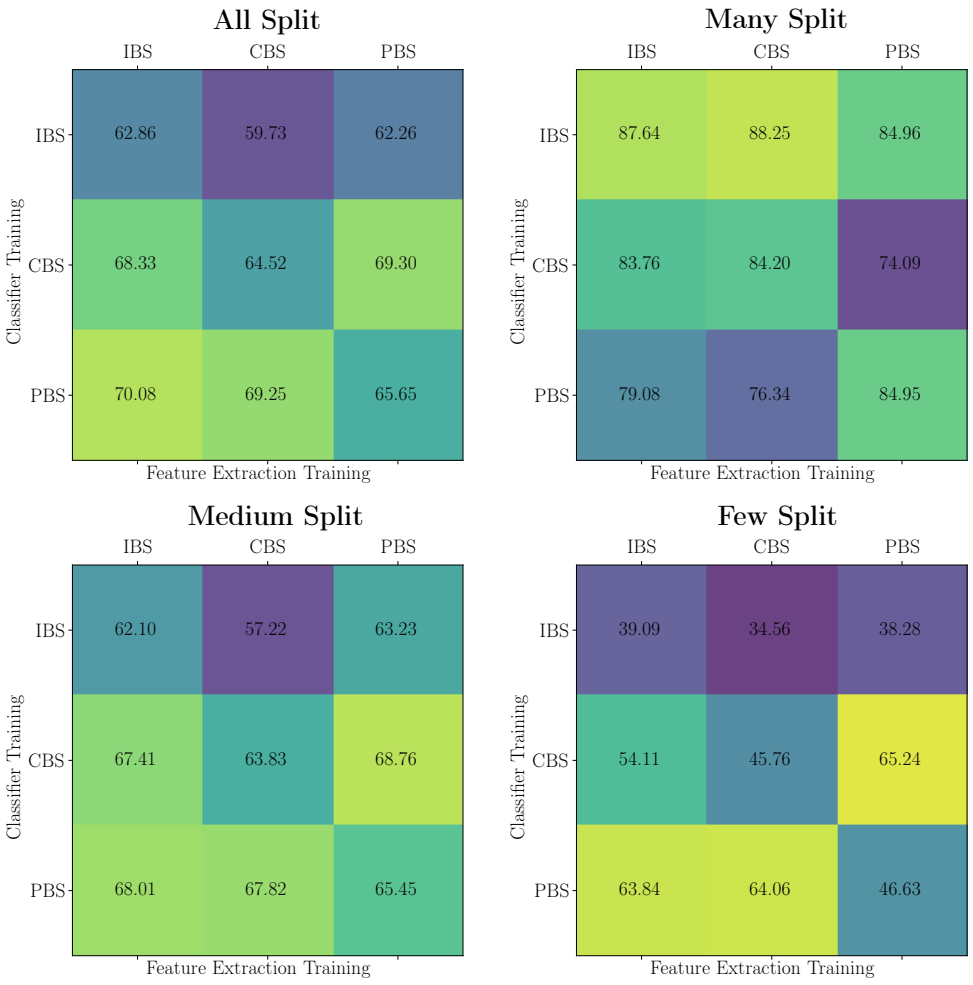


Figure 6.1: The performance for different ResNet-34 networks for each split on CIFAR-10 with $\rho = 50$. Different sampling schemes are used for feature learning and classifier learning: instance-balanced sampling (IBS), class-balanced sampling (CBS) and progressive-balanced sampling (PBS). As observed, when fixing the strategy for feature learning (comparing accuracy of three blocks in the vertical direction), the accuracy of classifiers trained with IBS are reasonably lower than CBS or PBS. When fixing the strategy for classifier training (comparing accuracies in the horizontal direction), the features trained with IBS or PBS perform better than those of CBS. The best performing method trains the feature extractor with PBS, and re-trains the classifier with CBS.

ing procedure into two parts: training the feature extractor and balancing the classifier. We apply three strategies each for feature learning and classifier learning, to obtain results for nine different combinations.

Figure 6.1 displays the results of decoupled training. Each grid in the figure represents the accuracy results for a specific class split. Combinations of sampling methods are arranged with the initial feature learning stage indi-

cated along the horizontal axis, and the subsequent classifier re-training stage indicated along the vertical axis.

The top left grid in Figure 6.1 displays the results among the all class split. The middle left score within this grid, for example, reports the averaged score of a network trained by an IBS-CBS procedure: feature learning under an IBS strategy, after which the classifier is re-trained under a CBS strategy. Reference to other decoupled sampling procedures will follow this nomenclature.

The results indicate that separately training the feature extractor and classifier may produce better results than training them jointly. Regardless of the learned features, training the classifier with an IBS strategy provides no noticeable benefits, while balancing the classifier with CBS or PBS improves generalisation. As the final stages of PBS transitions to CBS, this confirms that the classifier should be balanced in some way for better performance.

Furthermore, it seems that the most effective features are learned with either an IBS or a PBS strategy. Kang *et al.* (2020) recommend following an IBS procedure, but we find that PBS may offer equivalent or better performance. Our results may differ from theirs because of experimental differences such as training duration, learning rate decay and the added remix regularisation. When remix labels certain images, the distribution of the training dataset is inadvertently over-sampled for the minority, and could affect both the representation and classification learning stages.

For our experimental setup, the best performing procedure is a PBS-CBS strategy. Although IBS-PBS offers comparable results, with a higher score on the all and many class splits, the PBS-CBS strategy manages to provide the most similar scores between splits and offers the best results among the medium and few class splits, which might be desired for learning from class imbalanced data.

Figure 6.2 shows calibration results for the decoupled PBS-CBS procedure. The procedure substantially improves model calibration over all of the class splits. The initial extreme miscalibration in the few class splits (shown in Figure 4.3) is almost completely rectified, with accuracy nearly in line with averaged confidence. Additionally, the distributions of predicted probabilities are visually aligned across different class splits, with all confidences in the relatively high (0.8 to 1.0) range.

A notable disadvantage of decoupled sampling is the additional training needed to adjust the classifier after the features have been frozen. For very large datasets, over-sampling may not be feasible, or in some cases not possible. As an example, in object detection tasks it may not be practical to over-sample

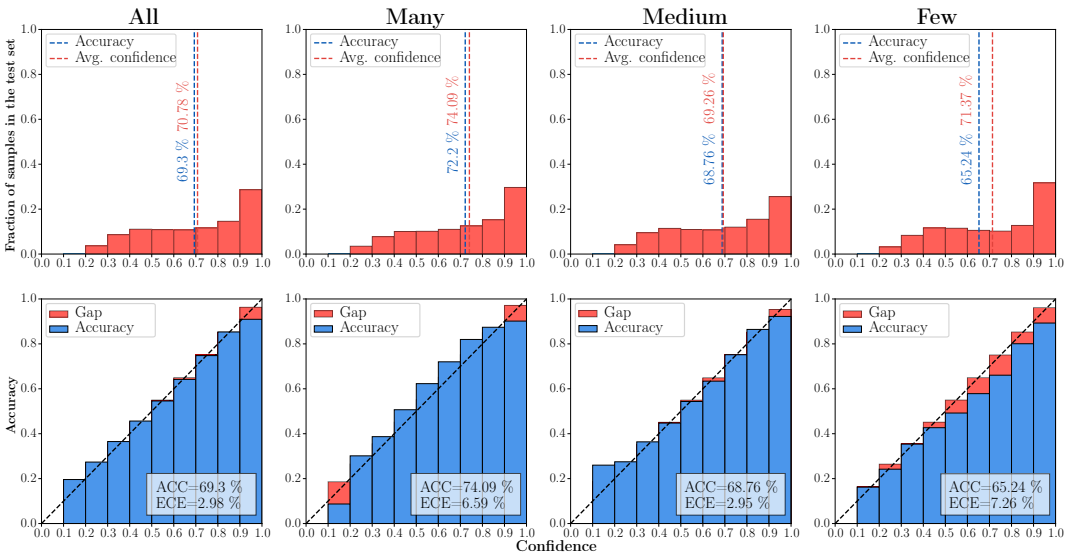


Figure 6.2: Calibration metrics for different class splits for the proposed decoupled PBS-CBS method. This approach achieves near optimal calibration.

minority instances without introducing additional majority instances as well (Wu *et al.*, 2020). Nevertheless, for our use case, the method of decoupled sampling is appropriate.

Methods closely related to this two-stage training scheme are briefly discussed next. Kang *et al.* (2020) propose to directly adjust the classifier weight norms following a normalisation procedure. The procedure scales classifier weights following the L_2 norm and controls the magnitude of the normalisation with an additional hyperparameter. This procedure offers similar performance to the decoupled sampling strategy while mitigating the need for additional network training.

Cao *et al.* (2019) find that decreasing the learning rate at the final stages of IBS training and adopting class-balanced sampling provide marked gains. Their method, referred to as deferred re-sampling (DRS), aims to decrease the extent of updating the feature extractor when switching to CBS, allowing only the classifier to be influenced by balanced updates.

Zhou *et al.* (2020) propose a bilateral-branch network that jointly trains the feature extractor and classifier with opposing sampling strategies. Following a cumulative learning procedure, the feature extractor first learns to model the imbalanced distribution, while the classifier is encouraged to model the test dataset distribution via a reverse-balanced sampling strategy.

6.2 Comparison of algorithmic methods

In this section we compare various methods in order to determine an effective strategy for learning from imbalanced data. The performance of these methods will be evaluated on the measured accuracy and calibration results across class splits. From the results in this section we wish to ascertain an appropriate approach for the class imbalance problem.

Unless indicated otherwise, the experimental setup for the model and dataset construction follows that of Section 2.2. Comparable methods are trained for the same number of mini-batch updates, apart from the decoupled and deferred sampling strategies where the classifier is trained for an additional 6,500 steps. Network parameters that achieved the lowest validation loss are reinstated as the final model. Results are obtained by averaging a total of 15 training sessions for each method. These sessions are produced by training on three randomly generated subsets of a particular CIFAR dataset, each trained five times. To simplify comparisons, we omit showing the performance of some methods, as their results are in line with others.

For methods that utilise mixup or remix, the hyperparameter setup is the same as in Chapter 5. They are set as $\alpha = 1$, $\kappa = 3$ and $\tau = 0.5$, where applicable. For SMOTE, the nearest neighbour hyperparameter is set to 5, following the default recommendation (Bowyer *et al.*, 2002). Additionally, we follow the proposed strategy of Cao *et al.* (2019) by combining the LDAM and DRS procedures, with their hyperparameter M tuned to normalise the margins so that the largest margin is 0.5, and decreasing the learning rate in the second stage of DRS training to 0.001.

Test accuracy results for different class splits are reported in Table 6.1, with highest accuracies per column in bold. Of the considered methods, the PBS-CBS procedure achieves the best results, with almost uniform scores across the different splits. While other methods achieve higher accuracies for the many class split, PBS-CBS manages to improve only upon the baseline. Moreover, all decoupled sampling methods (with the exception of PBS-PBS) outperform standard sampling methods on the few class split, demonstrating the surprising effect of re-training the classifier.

Comparing the performance of SMOTE with the CBS strategies highlights the benefit of incorporating mixup or remix regularisation. Both SMOTE and CBS over-sample the dataset until parity is achieved, and improves over the baseline. However, with no added regularisation (indicated as CBS), only a slight improvement over the baseline is achieved. Furthermore, the regularisation technique of SMOTE does not manage to offer the same gains as with mixup or remix. The increased performance over SMOTE could be because

CIFAR-10 ($\rho = 50$)				
Method	All Classes	Many Split	Medium Split	Few Split
IBS	38.6 ± 1.0	73.8 ± 0.9	31.9 ± 3.4	12.1 ± 4.2
IBS remix	62.9 ± 0.4	**86.8 ± 0.2**	63.3 ± 2.8	38.3 ± 6.9
CBS	39.8 ± 0.1	61.2 ± 2.7	40.4 ± 2.5	17.8 ± 1.1
SMOTE	55.2 ± 0.0	84.4 ± 0.5	50.5 ± 0.4	32.6 ± 1.4
CBS mixup	63.5 ± 0.2	79.4 ± 2.3	65.1 ± 2.6	45.3 ± 6.9
CBS remix	63.9 ± 0.1	80.6 ± 3.3	64.3 ± 2.0	46.8 ± 6.2
LDAM-DRS mixup	65.2 ± 1.3	72.3 ± 4.2	65.5 ± 6.6	63.2 ± 10.3
PBS-CBS remix	**69.3 ± 0.9**	74.0 ± 2.2	**68.9 ± 3.0**	**69.7 ± 5.2**

Table 6.1: The accuracy performance for different sampling methods for each split on CIFAR-10 with $\rho = 50$. Decoupled PBS-CBS with remix gives the best results.

mixup and remix sample image pairs from different classes as opposed to only sampling from the same class. This, coupled with label smoothing, could be what provides the superior performance.

Remix manages to improve upon mixup when a CBS strategy is used. Similar to the observations in Chapter 5, samples that satisfy the remix criteria will benefit generalisation on the minority classes, but at a cost of performance on the majority classes. Intuitively, using a CBS strategy should increase the effect of remix, but as can be seen, the improvements are not substantial over mixup training.

We consider PBS-CBS remix as an alternative to the LDAM-DRS method. To ensure a fair comparison, mixup is included with LDAM-DRS and both methods are trained for the same number of additional steps. The results show that adjusting margins indirectly with remix, while separately training the classifier may provide additional gains over LDAM-DRS. It is unclear however which of these strategies provide the greatest benefit over LDAM-DRS and we leave this for future work.

To compare the performance of these methods on a dataset other than CIFAR-10, Table 6.2 shows the result after training on an imbalanced subset of CIFAR-100 with $\rho = 50$. Recall that CIFAR-100 is a more difficult classification task than CIFAR-10, with more classes and fewer samples for each class. It is clear that the decoupled method offers superior gains. We believe it may be possible that different distributions and different datasets may warrant different decoupled sampling procedures, hence we also train a decoupled IBS-CBS sampling strategy. PBS-CBS outperforms IBS-CBS as well, with a slightly smaller gain for the many class split.

In Figure 6.3 the calibration metrics of models trained on the CIFAR-100

CIFAR-100 ($\rho = 50$)				
Method	All Classes	Many Split	Medium Split	Few Split
Baseline	23.4 ± 0.1	42.3 ± 0.2	21.3 ± 0.5	6.8 ± 0.5
IBS remix	30.3 ± 0.1	49.8 ± 0.2	28.7 ± 0.3	12.6 ± 0.1
CBS remix	29.5 ± 0.2	45.4 ± 0.8	28.7 ± 1.5	14.7 ± 2.2
LDAM-DRS mixup	34.1 ± 0.4	**51.1 ± 1.5**	33.3 ± 1.4	18.1 ± 2.3
IBS-CBS remix	33.2 ± 1.6	33.7 ± 8.5	35.1 ± 0.5	30.8 ± 5.1
PBS-CBS remix	**34.9 ± 1.3**	33.5 ± 4.2	**37.7 ± 0.1**	**33.3 ± 4.1**

Table 6.2: The accuracy performance for different sampling methods for each split on CIFAR-100 with $\rho = 50$. Decoupled PBS-CBS with remix gives the best results.

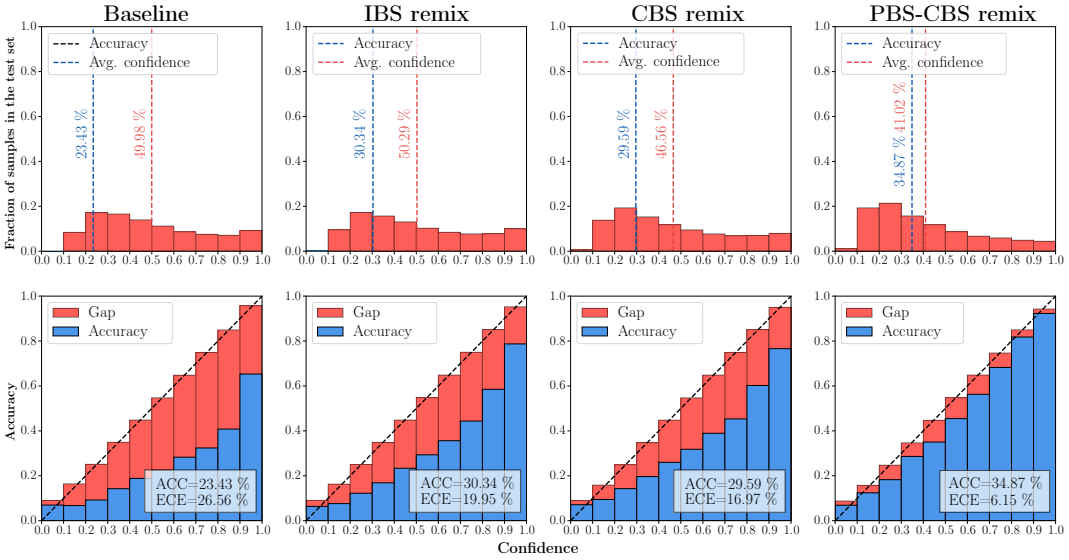


Figure 6.3: Performance comparison of model calibration between different strategies on CIFAR-100 with $\rho = 50$. The baseline implements no augmentation or sampling procedure. All methods improve both calibration and generalisation. Decoupled PBS-CBS produces the best calibration and generalisation. Methods such as SMOTE and IBS-CBS are not shown, as their results are in line with the others.

dataset (with $\rho = 50$) are shown for various methods. As can be seen, training the network with a CBS strategy improves upon standard IBS training, while PBS-CBS offers the best calibration results. The other methods are omitted as their results are in line with those shown.

Finally, Figure 6.4 shows the classifier weight norms of the baseline and PBS-CBS method when trained on the CIFAR-100 dataset with $\rho = 50$. Surprisingly, the weight norms are not uniform, with those of the minority classes noticeably higher than the majority. Intuitively this means that the model has learned to prioritise the minority classes.

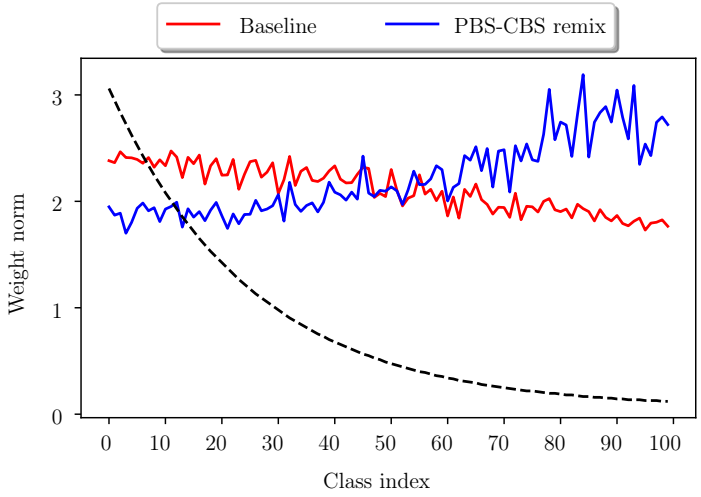


Figure 6.4: Per-class L_2 weight norms of the classification layer in a ResNet-34 model trained on an imbalanced CIFAR-100 dataset, with $\rho = 50$. Red: baseline model trained with weak augmentation. Blue: proposed decoupled PBS-CBS method with remix regularisation. Class distribution is super-imposed as a dashed line for reference. PBS-CBS with remix leads to higher weight norms for the minority, and lower for the majority, indicating that the classifier is more sensitive towards features from the minority classes.

The decoupled sampling strategy demonstrates considerable effectiveness for calibrating networks. It seems that for learning from class imbalanced data, network calibration is sensitive to both the classification layer and the learned representations produced by the network. This observation is in line with the work of Guo *et al.* (2017) who propose a network re-calibration method which bares similar resemblance with the decoupled sampling strategy.

In summary, re-sampling methods adjust network predictions to account for the imbalanced distribution of the training data. Over-sampling techniques, such as class-balanced sampling, manage to improve the generalisation of minority classes. These methods may however cause the network to overfit on duplicated samples as well as learn sub-optimal representations. To address these issues, we incorporate remix regularisation to reduce the effects of overfitting and demonstrate its effectiveness over classical methods such as SMOTE. To improve representation learning we train the network architecture in a decoupled manner following a PBS-CBS scheme.

The methods presented in this chapter manage to improve both network generalisation and calibration, with the proposed PBS-CBS remix technique offering the greatest benefit. In the next chapter we consider including additional data with semi-supervised methods to construct more class balanced datasets for

training. The proposed sampling procedure developed in this chapter will show to be advantageous in this learning scheme, and will form part of the final approach to produce effective models.

Chapter 7

Semi-supervised learning methods

As the barriers of data accessibility are eased with modern technology, a possible solution to the class imbalance problem is to train a network on a more balanced dataset, supplemented with additional data. Curating additional data can however be costly if specific domain knowledge is required for labelling. For example, medical data may require a skilled professional to provide a ground truth label for each sample.

Semi-supervised methods are often considered as a means to circumvent the need for human labelling. Semi-supervised learning is concerned with improving model performance when learning from scarcely populated data, by exploiting large amounts of additional unlabelled data. For image classification there are two primary approaches: consistency regularisation and pseudo labelling.

Consistency regularisation methods aim to alleviate the need for the unlabelled data to be labelled when training. These methods train networks to predict similar outputs for images that have been subjected to different semantic-preserving perturbations or augmentations (Liu and Tan, 2021; Sajjadi *et al.*, 2016). Pseudo labelling methods attempt to label the unlabelled data, allowing a network to further train on the newly introduced data in a supervised learning manner (Cascante-Bonilla *et al.*, 2020; Xie *et al.*, 2020; Lee, 2013). We focus further on pseudo labelling for improving model generalisation when learning from class imbalanced data.

7.1 Pseudo labelling

Pseudo labelling posits that under the cluster assumption (where each class occupies a distinct cluster in latent feature space), unlabelled sample features

that reside within a high density cluster should likely be from the same class (Chapelle and Zien, 2005). As such, pseudo labelling methods artificially label unlabelled samples according to the confidence prediction made by some learned network. These pseudo samples are then treated as true observations in subsequent training iterations (Lee, 2013).

However, as illustrated in Figure 1.3, when learning from class imbalanced data the network may bias towards majority classes and produce a decision boundary that passes through the latent feature space of minority clusters. As a result, unlabelled samples within minority clusters may be incorrectly pseudo labelled (Hyun *et al.*, 2020).

Additionally, semi-supervised methods are known to be susceptible to confirmation bias (Arazo *et al.*, 2020). Confirmation bias results from repeatedly using incorrect predictions on unlabelled data during training, thereby increasing the network's confidence in incorrect predictions, and producing a network that will tend to resist corrections or changes. For pseudo labelling, confirmation bias manifests when images are incorrectly labelled and included in the training set which propagates to subsequent training stages. Coupled with incorrect boundaries that pass through minority clusters in feature space, this could deleteriously affect the pseudo learning process (Hyun *et al.*, 2020).

To investigate the effects of confirmation bias when learning from class imbalanced data, we proceed to determine how model bias may affect each stage of the pseudo labelling process. Specifically, we first determine several differences of constructed pseudo datasets when a network is trained on class imbalanced data, and then determine how this may further affect training, and whether an appropriate solution may alleviate observed bias.

Following Xie *et al.* (2020), a noisy student pseudo labelling framework is implemented for experimentation. The method simplifies to an iterative process, described as follows.

1. A teacher network $f(\theta_t)$ is trained on the originally labelled dataset D_L until convergence is observed.

2. The teacher network produces confidence predictions on the unlabelled dataset D_U, pseudo labelling each sample with the class that achieves the highest confidence score. Pseudo labelled samples with confidence scores exceeding some predetermined confidence threshold α are then used to construct a pseudo dataset D_P.

3. A re-initialised student network $f(\theta_s)$ is trained on both D_L and D_P.

4. The process repeats from step 2 with the student network replacing the teacher network, until convergence is observed.

In general, it would be desirable to learn from as much correctly labelled data as possible. However as network predictions may be incorrect, the teacher network may include significant amounts of incorrectly labelled data. In response, a confidence threshold is introduced to act as a filter to allow only sufficiently confident predictions into D_P.

On the other hand, highly confident predictions may introduce a disproportionate number of samples that reside within high density clusters, far from the decision boundary. Hence relatively uninformative samples may dominate the training process, and subdue informative samples needed for effective learning. Consequently, a modified loss is typically introduced during student network training to prioritise samples from D_L. The modified loss is calculated as

$$L_s = L(D_L, \theta) + \nu L(D_P, \theta), \tag{7.1}$$

where the weight ν is a hyperparameter (Lee, 2013).

Following Xie *et al.* (2020), when training the student network, the dataset is subjected to regularisation techniques, while when generating the pseudo labels, regularisation is removed. In effect, samples the teacher is confident about will be introduced as more difficult variations (noise) for the student network to be consistent with. Remix and weak augmentation will be used as regularisation methods in our experiments.

The choice of ν and α may be highly dependent on D_L and D_U. For simplicity, we follow the settings used by Sohn *et al.* (2020) and set ν to 1, effectively combining both datasets equally for training. We set α to 95%, meaning all samples regardless of class should produce confidence prediction above 95% to be included in the pseudo dataset. For a perfectly calibrated model, it should then be expected that the proportion of incorrectly labelled samples included within the pseudo dataset is close to 5%.

7.2 Teacher networks

We proceed to demonstrate the effects of the teacher network when constructing D_P by different learning methods. To compare between networks, several key characteristics for D_P are first identified.

Yang and Xu (2020) determine that dataset characteristics such as the class

domain (representation of the underlying classes), class distribution and size of the pseudo dataset are important for improving the performance of the student network. In general, they recommend the pseudo dataset to be at least as large as D_L and to be less imbalanced than the labelled dataset. Furthermore, they demonstrate that as the degree of class domain overlap reduces between datasets, the performance of the network deteriorates. In addition, we will measure the amount of incorrectly labelled data within the pseudo datasets, denoted as corruption.

As D_P is constructed from unlabelled data, the characteristics of D_U should therefore be taken into consideration. To meet the recommendations of Yang and Xu (2020) it could be assumed that D_U should be larger and consist of a similar class domain to that of D_L. In practice, however, these characteristics of D_U are not known and difficult to determine. Instead D_U is typically constructed from the same or similar domain source as was used to construct D_L (Sohn *et al.*, 2020; Carlini *et al.*, 2019; Oliver *et al.*, 2018).

For our experiments, CIFAR-10 (with $\rho = 50$) is used as D_L and the CINIC-10 dataset as D_U. CINIC-10 is a balanced dataset that is constructed from a different yet similar domain to CIFAR-10, consisting of the same classes, but is much larger than CIFAR-10 (Darlow *et al.*, 2018). This makes CINIC-10 a suitable choice as an unlabelled dataset. To determine the amount of corruption in the pseudo dataset, the true labels of CINIC-10 are compared with the predicted labels from the teacher network. During training the original labels of CINIC-10 are ignored. Further details of CINIC-10 are given in Section 2.1.

Table 7.1 reports the size, balance and corruption between datasets that are constructed with different teacher networks. The first entry, D_L, serves as reference and is the original labelled dataset. The others stem from combining D_L with D_P. The first dataset, D_{P_1}, is obtained with $f(\theta_t)$ trained with an IBS-aug scheme, while the second dataset, D_{P_2}, is produced from following the PBS-CBS remix scheme.

Recall that the IBS-aug scheme should reflect a standard training procedure that does not account for class imbalance or network calibration. When training the student network with the PBS-CBS strategy, in the second decoupled stage we balance the classifier only on D_L, circumventing the computational cost of training on large amounts of over-sampled pseudo data.

Both D_{P_1} and D_{P_2} are larger than D_L, with D_{P_1} containing significantly more data for training than D_{P_2}. From the work of Yang and Xu (2020) it can be said that the larger pseudo datasets should improve further training. However, D_{P_1} contains considerable levels of corruption and struggles to improve upon the class distribution of D_U. This indicates that the biased confidence

Name	Algorithm	Size	Imbalance (ρ)	Corruption (%)
D_L	None	11,100	50	0.0
D_{P_1}	IBS-aug	120,505 ± 4,169	48.8 ± 27.4	41.4 ± 3.1
D_{P_2}	PBS-CBS remix	35,104 ± 4,969	21.0 ± 6.2	7.2 ± 0.3

Table 7.1: Differences between pseudo datasets when a teacher network is trained with different methods. D_{P_1} is produced from standard IBS training with weak augmentation. D_{P_2} is produced from the decoupled PBS-CBS method. When a teacher is better calibrated and accounts for class imbalanced data, the pseudo dataset is smaller, contains less incorrectly labelled data and is more class balanced. For reference, the original labelled dataset D_L is included.

and generalisation from the teacher has been transferred to the pseudo dataset.

In contrast, the PBS-CBS remix strategy improves dataset characteristics, with D_{P_2} producing better class distributions and lower corruption levels. Furthermore, as the network is better calibrated and less prone to make overconfident predictions, fewer samples are introduced, lowering the dataset size significantly.

On closer inspection, it seems that because the unlabelled dataset is class balanced, the PBS-CBS procedure should be expected to produce balanced class distributions as well. It could be argued that the network struggles to classify predictions in line with CIFAR-10, as samples from CINIC-10 are not constructed from the same domain as that of CIFAR-10. Moreover, the amount of corruption seems to be in line with what should be expected from the given confidence threshold and the method's ECE score.

Regardless, the comparison between these datasets shows that the bias from learning with class imbalanced data will affect the quality of the pseudo dataset.

7.3 Student networks

We proceed to determine how sensitive the student network is to confirmation bias, by training on the different pseudo datasets.

Table 7.2 displays the class split performance for various configurations. The algorithm column indicates the learning strategy used for training the student network, and the dataset column indicates the dataset the network was train on. We compare the two strategies from the previous section, namely IBS with weak augmentation and the proposed PBS-CBS with remix regularisation. As

Algorithm	Dataset	All	Many	Medium	Few
IBS-aug	D_L	60.7 ± 0.1	**87.5 ± 0.1**	58.5 ± 1.8	37.0 ± 2.9
IBS-aug	D_{P_1}	58.0 ± 2.6	84.8 ± 3.1	56.1 ± 8.4	33.7 ± 5.9
IBS-aug	D_{P_2}	62.5 ± 2.5	70.1 ± 6.1	60.5 ± 5.0	57.6 ± 8.7
PBS-CBS remix	D_L	69.3 ± 0.8	74.1 ± 1.5	68.8 ± 5.5	65.2 ± 7.2
PBS-CBS remix	D_{P_2}	**73.4 ± 1.8**	77.6 ± 1.5	**70.2 ± 3.8**	**72.3 ± 5.2**

Table 7.2: Measured accuracies across class splits for student networks trained on different datasets. Standard IBS training with weak augmentation performs best when trained on the D_{P_2} dataset. Training the student network with the decoupled PBS-CBS method outperforms the best supervised (D_L) results.

the datasets are far larger, mini-batch updates are increased to 30,000 during training.

Results from training with the same IBS strategy on the different datasets indicate that the student network is sensitive to the teacher network used. When trained on D_{P_1}, for example, the network struggles to improve over the performance of the teacher, with no clear improvements across the different class splits. This indicates that the student training was unsuccessful and additional iterative training might further deteriorate network performance. Moreover, the results have not changed between the teacher and student network, meaning that the network still displays a clear inclination towards the majority classes at the expense of the minority. Hence it seems that not only has the bias from the teacher been transferred to the student network, but iterative training may also deteriorate performance.

When trained with the IBS strategy on D_{P_2}, the model manages to produce more uniform results, with a noticeable increase on the generalisation of the minority classes. The increase in generalisation of the minority classes could stem from the increased number of minority samples for training overall, but as D_{P_1} contains significantly more samples, it seems that the number of correctly labelled samples and severity of the class distribution are of importance.

Even though the number of majority samples is increased, the student network displays diminished gains on majority classes over the teacher network. The reduced performance could be as a result of the network training on a more balanced dataset. As shown in Figure 1.4, training on a more balanced dataset may reduce generalisation on the majority classes as the network's tendency to favour these classes is lessened. Furthermore, the student network manages to improve over the teacher network, indicating that additional performance could be obtained if the pseudo labelling process is repeated.

Finally, we train the student network with the PBS-CBS scheme on D_{P_2}. Ta-

ble 7.2 shows that when both the teacher and student networks are trained to account for class imbalance and model calibration, we are able to obtain improved performance over the best performing supervised training.

Our choice of D_U could be a misrepresentation of realistic class imbalanced semi-supervised settings, as the unlabelled dataset could naturally be class imbalanced as well. Yang and Xu (2020) found that the distribution of the unlabelled dataset will influence the performance of learning from imbalanced data. With fewer samples in minority classes available for pseudo labelling, a class imbalanced D_U will increase the difficulty of producing sufficiently balanced pseudo datasets, which will increase the difficulty of the learning process.

To investigate, the CINIC-10 dataset is artificially imbalanced following the same distribution as CIFAR-10 with $\rho = 50$, and the pseudo labelling procedure is re-implemented. For this particular class imbalance, D_U contains only 360 samples of the smallest minority class and, as a result, the pseudo dataset may exhibit a more imbalanced distribution than D_L. To prohibit the student network from training on a more severely class imbalanced dataset, samples from majority classes are removed from the pseudo dataset until a class imbalance of at least $\rho = 50$ is achieved. To determine which samples to remove, the samples are ordered according to the confidence scores from the teacher network and the least confident samples are progressively omitted. This additional step was not needed for the PBS-CBS procedure when training with a class balanced D_U.

Table 7.3 compares the dataset characteristics following the PBS-CBS strategy for teacher network training on D_L. As expected, dataset D_{P_3} consists of far fewer pseudo labelled samples and is more class imbalanced than D_{P_2}. Surprisingly, the amount of corruption within the dataset is higher. It is unclear whether the increased corruption is caused by covariate shift when constructing imbalanced datasets of D_U or from systematically removing low confident samples. Regardless, it should be noted that the ECE score does not reflect the expected amount of corruption of the network as was the case with a balanced dataset, and the higher levels of corruption should increase the learning difficulty in subsequent training sessions.

Table 7.4 reports the student performance and shows that generalisation on minority classes deteriorates, likely from the lack of additional minority samples and additional corruption. As a result, the network fails to improve over the supervised learning baseline, and additional training may further deteriorate network learning.

Determining an appropriate value for the confidence threshold seems to be an important consideration. The confidence threshold indirectly determines the

Name	Algorithm	Size	Imbalance (ρ)	Corruption (%)
D_L	None	11,100	50	0.0
D_{P_2}	PBS-CBS remix	35,104 ± 4,969	21.0 ± 6.2	7.2 ± 0.3
D_{P_3}	PBS-CBS remix	13,528 ± 1724	38.8 ± 12.2	10.0 ± 0.05

Table 7.3: Dataset characteristics of the pseudo labelling process when the unlabelled dataset D_U follows the same class distribution as the labelled dataset D_L. D_P is the training dataset after constructing the pseudo dataset. Pseudo labelling with a severely class imbalanced D_{P_3} reduces the number of samples for training, while increasing corruption and class imbalance levels of the pseudo dataset. D_{P_2} is constructed from a class balanced D_U for comparison.

Algorithm	Dataset	All	Many	Medium	Few
PBS-CBS remix	D_L	69.3 ± 0.1	74.1 ± 1.5	68.8 ± 5.5	65.2 ± 7.2
PBS-CBS remix	D_{P_2}	**73.4 ± 1.8**	77.6 ± 1.5	**70.2 ± 3.8**	**72.3 ± 5.2**
PBS-CBS remix	D_{P_3}	69.4 ± 1.2	**81.6 ± 1.0**	67.6 ± 4.2	59.5 ± 4.9

Table 7.4: Network performance of the pseudo labelling process when the unlabelled dataset D_U follows the same class distribution as the labelled dataset D_L. D_P is the training dataset after constructing the pseudo dataset. When learning on a severely class imbalanced dataset D_{P_3}, the network struggles to improve over the supervised learning benchmark D_L. D_{P_2} is constructed from a class balanced D_U for comparison.

number of corrupted samples within the pseudo dataset, but as generalisation and network calibration may be dependent on the distribution of the training data, and typically underperforms on minority classes, a global threshold could fail to appropriately filter the unlabelled data. Adjusting a confidence threshold to account for the distribution could be a solution to produce additional pseudo samples for the minority, but since the calibration of minority classes are most likely miscalibrated, this could produce higher proportions of corrupted samples.

Most semi-supervised methods are developed under the assumption that datasets are class balanced. Only recently has the task of learning from class imbalanced data in semi-supervised learning been explored, and it remains a relatively new research direction for deep learning. For completeness, we highlight exploratory methods recently proposed. Kim *et al.* (2020) investigate pseudo labelling tasks where both D_L and D_U are class imbalanced. They note that learning from class imbalanced data will produce samples that are not representative of the true class distribution of the unlabelled dataset, and propose distribution aligning refinery for pseudo labelling (DARP). Their method aims to adjust the pseudo labelled samples so that the distribution matches the distribution of the unlabelled dataset through a convex optimisation problem.

Hyun *et al.* (2020) investigate consistency regularisation based semi-supervised methods for class imbalanced learning. They find that the performance deterioration on minority classes in consistency regularisation based methods are as a result firstly of the decision boundary crossing minority clusters in latent feature space, and secondly that consistency regularisation will smooth the decision boundary at these points. Consequently, the smoothed boundary will increase the number of misclassified minority classes, and degrade performance. They propose a suppressed consistency loss (SCL) that reduces the severity of regularisation for samples near these boundaries.

Overall, the results demonstrate the significance of producing networks that are well-calibrated across different class frequencies and produce sufficient amounts of pseudo labelled data for each class. Furthermore, provided the unlabelled dataset is closely related to the labelled dataset (within the same domain) and sufficiently large, confirmation bias in pseudo labelling methods can be mitigated when learning from class imbalanced datasets, and improve performance over supervised learning methods. However, the number of unlabelled minority samples within the unlabelled dataset seems to be crucial for effective learning and in practical semi-supervised applications, where the unlabelled dataset may be class imbalanced, this may not be possible to obtain.

Chapter 8

Conclusions

We studied several effects when convolutional neural networks are trained on class imbalanced data. Based on our findings, both supervised and semi-supervised solutions were proposed.

Our investigations revealed that when learning from class imbalanced data, minority classes may not be representative of their underlying distributions, resulting in the decision boundary shifting into latent feature space occupied by these classes. When training neural networks, we showed that the weight norms of the classification layer may skew according to the distribution of the training data, and further demonstrated that the confidence scores of network predictions are likely to not only be overconfident, but biased towards majority classes. To address these observations, we proposed a margin based regularisation technique to indirectly increase the margins of minority classes in order to shift the decision boundary and improve network calibration. To adjust the classifier weight norms, we proposed a decoupled sampling technique to allow the network to learn the feature representations and classifier more efficiently.

Our experiments shed light on the importance of incorporating effective regularisation methods for learning from class imbalanced data. While semantic-preserving augmentations such as image translations provide considerable improvements towards generalisation, sample interpolations can be more effective at improving confidence calibration. Remix regularisation manages to shift the network's decision boundary by increasing minority class margins, however the method necessitates additional hyperparameter tuning and offers marginal improvements over standard mixup regularisation. In general, the regularisation methods are easy to implement, computationally efficient and can be incorporated with other methods.

For convolutional neural networks, training the feature extractor and classifier separately illustrates the difficulties of applying re-sampling techniques for the class imbalance problem. On the one hand, adjusting the weight norms of the

classifier following some class balancing scheme increases network performance, but this may be at the expense of the learned representations. We found that networks tend to produce more generalisable representations when the feature extractor is initially trained following the original imbalanced training distribution, which counteracts traditional re-sampling methods. Our proposed decoupling scheme offers improved performance over these traditional methods while remaining simple to construct, requires no hyperparameter tuning and can be easily incorporated with other methods.

The methods we considered manage to offer improvements over standard training techniques, however we believe they can be improved upon. Possible areas for improvement entail devising distribution-aware sampling strategies for image interpolations, optimising the remix parameters over separate classes, and mitigating the need for additional training during decoupled sample training.

For semi-supervised learning, network performance were found to be sensitive to characteristics of both the labelled and unlabelled datasets. We found that pseudo labelling methods, reliant upon both accurate and well-calibrated predictions, are likely to inherit class imbalanced bias and may deteriorate network performance in subsequent training stages. Combining our decoupled sampling method with remix regularisation manages to reduce network bias and produces more balanced and less corrupted pseudo labelled datasets than standard network training. However, in settings where the unlabelled dataset shares the same imbalanced distribution as the labelled data, our proposed method struggles to provide a sufficient number of correctly labelled pseudo samples for training and subsequently fails to improve generalisation of minority classes.

Our exploration into the utility of semi-supervised methods demonstrated the importance of incorporating appropriate unlabelled data. We found that characteristics such as the size and distribution of the dataset may influence network performance, however in practical settings these characteristics are typically not known. Moreover, as unlabelled datasets tend to be far larger than the labelled datasets, training may be expensive. As such, potential future work should investigate methods to determine these characteristics preemtively.

Overall, our work demonstrates that network performance on minority classes can be improved with general solutions. However, the datasets we considered are curated for academic research, and the efficacy of our methods were tested with accuracy and calibration metrics alone. In real world applications such as medical diagnosis or autonomous driving, the data may impose ethical constraints on the learning process, including fairness or privacy concerns. As such, it remains paramount to appropriately consider potential risks of pro-

ducing unfair or biased results in critical, high-stake applications.

References

Abadi, M., Barham, P., Chen, J., Chen, Z., Davis, A., Dean, J., Devin, M., Ghemawat, S., Irving, G., Isard, M., Kudlur, M., Levenberg, J., Monga, R., Moore, S., Murray, D.G., Steiner, B., Tucker, P.A., Vasudevan, V., Warden, P., Wicke, M., Yu, Y. and Zheng, X. (2016). TensorFlow: a system for large-scale machine learning. In: *12th USENIX Symposium on Operating Systems Design and Implementation, Savannah, GA, USA, November 2-4, 2016*, pp. 265–283. USENIX Association.
Available at: `http://arxiv.org/abs/1605.08695`

Arazo, E., Ortego, D., Albert, P., OâConnor, N.E. and McGuinness, K. (2020). Pseudo-labeling and confirmation bias in deep semi-supervised learning. In: *International Joint Conference on Neural Networks*, pp. 1–8.
Available at: `http://arxiv.org/abs/1908.02983`

Bartlett, P.L., Foster, D.J. and Telgarsky, M. (2017). Spectrally-normalized margin bounds for neural networks. *Conference on Neural Information Processing Systems*.
Available at: `http://arxiv.org/abs/1706.08498`

Bellinger, C., Corizzo, R. and Japkowicz, N. (2021). Calibrated resampling for imbalanced and long-tails in deep learning. In: *Discovery Science - 24th International Conference, Halifax, Canada*, vol. 12986 of *Lecture Notes in Computer Science*, pp. 242–252. Springer.
Available at: `https://doi.org/10.1007/978-3-030-88942-5_19`

Bowyer, K.W., Chawla, N.V., Hall, L.O. and Kegelmeyer, W.P. (2002). SMOTE: synthetic minority over-sampling technique. *Journal of Artificial Intelligence Research*, vol. 16, pp. 321–357.
Available at: `https://doi.org/10.1613/jair.953`

Branco, P., Torgo, L. and Ribeiro, R.P. (2015). A survey of predictive modelling under imbalanced distributions. *CoRR*, vol. abs/1505.01658. `1505.01658`.
Available at: `http://arxiv.org/abs/1505.01658`

Buda, M., Maki, A. and Mazurowski, M.A. (2018). A systematic study of the class imbalance problem in convolutional neural networks. *Neural Networks*, vol. 106, pp. 249–259.
Available at: `https://doi.org/10.1016/j.neunet.2018.07.011`

Cao, K., Wei, C., Gaidon, A., Aréchiga, N. and Ma, T. (2019). Learning imbalanced datasets with label-distribution-aware margin loss. In: *Advances in Neural Information Processing Systems: Annual Conference on Neural Information Processing Systems*, pp. 1565–1576.
Available at: `http://arxiv.org/abs/1906.07413`

Carlini, N., Erlingsson, Ú. and Papernot, N. (2019). Distribution density, tails, and outliers in machine learning: Metrics and applications. *CoRR*, vol. abs/1910.13427.
Available at: `http://arxiv.org/abs/1910.13427`

Carratino, L., Cissé, M., Jenatton, R. and Vert, J. (2020). On mixup regularization. *CoRR*, vol. abs/2006.06049. `2006.06049`.
Available at: `https://arxiv.org/abs/2006.06049`

Cascante-Bonilla, P., Tan, F., Qi, Y. and Ordonez, V. (2020). Curriculum labeling: Self-paced pseudo-labeling for semi-supervised learning. *CoRR*, vol. abs/2001.06001. `2001.06001`.
Available at: `https://arxiv.org/abs/2001.06001`

Chapelle, O. and Zien, A. (2005). Semi-supervised classification by low density separation. In: *Proceedings of the Tenth International Workshop on Artificial Intelligence and Statistics, Bridgetown, Barbados, January 6-8 2005*. Society for Artificial Intelligence and Statistics.
Available at: `http://www.gatsby.ucl.ac.uk/aistats/fullpapers/198.pdf`

Chou, H., Chang, S., Pan, J., Wei, W. and Juan, D. (2020). Remix: Rebalanced mixup. In: *Computer Vision - European Converence on Computer Vision 2020 Workshops, Glasgow, UK, August 23-28, 2020*, vol. 12540 of *Lecture Notes in Computer Science*, pp. 95–110. Springer.
Available at: `https://doi.org/10.1007/978-3-030-65414-6_9`

Cui, Y., Gu, Z., Mahajan, D., van der Maaten, L., Belongie, S.J. and Lim, S. (2019*a*). Measuring dataset granularity. *CoRR*, vol. abs/1912.10154. `1912.10154`.
Available at: `http://arxiv.org/abs/1912.10154`

Cui, Y., Jia, M., Lin, T., Song, Y. and Belongie, S.J. (2019*b*). Class-balanced loss based on effective number of samples. In: *Institute of Electrical and Electronics Engineers (IEEE) Conference on Computer Vision and Pattern Recognition, Long Beach, CA, USA, June 16-20, 2019*, pp. 9268–9277. Computer Vision Foundation / IEEE.
Available at: `http://arxiv.org/abs/1901.05555`

Darlow, L.N., Crowley, E.J., Antoniou, A. and Storkey, A.J. (2018). CINIC-10 is not ImageNet or CIFAR-10. *CoRR*, vol. abs/1810.03505. `1810.03505`.
Available at: `http://arxiv.org/abs/1810.03505`

Dedduwakumara, D.S. and Prendergast, L.A. (2020). Confidence intervals for quantiles from histograms and other grouped data. *Communications in Statistics -*

Simulation and Computation, vol. 49, no. 6, pp. 1546–1559.
Available at: https://doi.org/10.1080/03610918.2018.1499935

DeGroot, M.H. and Fienberg, S.E. (1983). The comparison and evaluation of forecasters. *Journal of the Royal Statistical Society. Series D (The Statistician)*, vol. 32, no. 1/2, pp. 12–22.
Available at: http://www.jstor.org/stable/2987588

Gu, J., Wang, Z., Kuen, J., Ma, L., Shahroudy, A., Shuai, B., Liu, T., Wang, X., Wang, G., Cai, J. and Chen, T. (2018). Recent advances in convolutional neural networks. *Pattern Recognition*, vol. 77, pp. 354–377.
Available at: https://doi.org/10.1016/j.patcog.2017.10.013

Guo, C., Pleiss, G., Sun, Y. and Weinberger, K.Q. (2017). On calibration of modern neural networks. In: *Proceedings of the 34th International Conference on Machine Learning, Sydney, NSW, Australia, August 6-11, 2017*, vol. 70 of *Proceedings of Machine Learning Research*, pp. 1321–1330. PMLR.
Available at: http://arxiv.org/abs/1706.04599

Guo, H., Mao, Y. and Zhang, R. (2019). Mixup as locally linear out-of-manifold regularization. In: *31st Innovative Applications of Artificial Intelligence Conference, The ninth Symposium on Educational Advances in Artificial Intelligence, Honolulu, Hawaii, USA, January 27 - February 1, 2019*, pp. 3714–3722. AAAI Press.
Available at: http://arxiv.org/abs/1809.02499

Hanson, S.J. and Pratt, L.Y. (1988). Comparing biases for minimal network construction with back-propagation. In: *Advances in Neural Information Processing Systems, NIPS Conference, Denver, Colorado, USA, 1988*, pp. 177–185. Morgan Kaufmann.
Available at: http://papers.nips.cc/paper/156-comparing-biases-for-minimal-network-construction-with-back-propagation

He, K., Zhang, X., Ren, S. and Sun, J. (2016). Deep residual learning for image recognition. In: *IEEE Conference on Computer Vision and Pattern Recognition, Las Vegas, NV, USA, June 27-30, 2016*, pp. 770–778. IEEE Computer Society.
Available at: http://arxiv.org/abs/1512.03385

Horn, G.V., Aodha, O.M., Song, Y., Shepard, A., Adam, H., Perona, P. and Belongie, S.J. (2017). The inaturalist challenge 2017 dataset. *CoRR*, vol. abs/1707.06642. 1707.06642.
Available at: http://arxiv.org/abs/1707.06642

Horn, G.V. and Perona, P. (2017). The devil is in the tails: Fine-grained classification in the wild. *CoRR*, vol. abs/1709.01450. 1709.01450.
Available at: http://arxiv.org/abs/1709.01450

Hyun, M., Jeong, J. and Kwak, N. (2020). Class-imbalanced semi-supervised learning. *CoRR*, vol. abs/2002.06815. 2002.06815.
Available at: https://arxiv.org/abs/2002.06815

Ioffe, S. and Szegedy, C. (2015). Batch normalization: Accelerating deep network training by reducing internal covariate shift. In: *Proceedings of the 32nd International Conference on Machine Learning*, vol. 37 of *ICML*, pp. 448 – 456. Journal of Machine Learning Research.
Available at: `http://arxiv.org/abs/1502.03167`

Japkowicz, N. and Stephen, S. (2002). The class imbalance problem: A systematic study. *Intelligent Data Analysis*, vol. 6, no. 5, pp. 429–449.
Available at: `https://doi.org/10.1016/j.neunet.2018.07.011`

Johnson, J. and Khoshgoftaar, T. (2019). Survey on deep learning with class imbalance. *Journal of Big Data*, vol. 6, p. 27.
Available at: `https://journalofbigdata.springeropen.com/articles/10.1186/s40537-019-0192-5`

Kang, B., Xie, S., Rohrbach, M., Yan, Z., Gordo, A., Feng, J. and Kalantidis, Y. (2020). Decoupling representation and classifier for long-tailed recognition. In: *8th International Conference on Learning Representations, Addis Ababa, Ethiopia, April 26-30, 2020*.
Available at: `http://arxiv.org/abs/1910.09217`

Kim, J., Hur, Y., Park, S., Yang, E., Hwang, S.J. and Shin, J. (2020). Distribution aligning refinery of pseudo-label for imbalanced semi-supervised learning. In: *Advances in Neural Information Processing Systems: Annual Conference on Neural Information Processing Systems 2020, December 6-12, 2020, virtual*.
Available at: `https://arxiv.org/abs/2007.08844`

Krawczyk, B. (2016). Learning from imbalanced data: open challenges and future directions. *Progress in Artificial Intelligence*, vol. 5, no. 4, pp. 221–232.
Available at: `https://doi.org/10.1007/s13748-016-0094-0`

Krizhevsky, A. and Hinton, G. (2009). Learning multiple layers of features from tiny images. Tech. Rep., University of Toronto, Toronto, Ontario.
Available at: `https://www.cs.toronto.edu/~kriz/cifar.html`

Kumar, A., Liang, P. and Ma, T. (2019). Verified uncertainty calibration. In: *Advances in Neural Information Processing Systems: Annual Conference on Neural Information Processing Systems, NeurIPS 2019, Vancouver, BC, Canada December 8-14, 2019,*, pp. 3787–3798.
Available at: `http://arxiv.org/abs/1909.10155`

Lee, D.-H. (2013). Pseudo-label : The simple and efficient semi-supervised learning method for deep neural networks. *International Conference on Machine Learning 2013 Workshop: Challenges in Representation Learning*.
Available at: `https://www.researchgate.net/publication/280581078_Pseudo-Label_The_Simple_and_Efficient_Semi-Supervised_Learning_Method_for_Deep_Neural_Networks`

Leevy, J.L., Khoshgoftaar, T.M., Bauder, R.A. and Seliya, N. (2018). A survey on addressing high-class imbalance in big data. *Journal of Big Data*, vol. 5, no. 1, pp. 1–30.
Available at: https://doi.org/10.1186/s40537-018-0151-6

Levinson, J., Askeland, J., Becker, J., Dolson, J., Held, D., Kammel, S., Kolter, J.Z., Langer, D., Pink, O., Pratt, V.R., Sokolsky, M., Stanek, G., Stavens, D.M., Teichman, A., Werling, M. and Thrun, S. (2011). Towards fully autonomous driving: Systems and algorithms. In: *IEEE Intelligent Vehicles Symposium (IV), Baden-Baden, Germany, June 5-9, 2011*, pp. 163–168. IEEE.
Available at: https://doi.org/10.1109/IVS.2011.5940562

Lin, T., Goyal, P., Girshick, R.B., He, K. and Dollár, P. (2020). Focal loss for dense object detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 42, no. 2, pp. 318–327.
Available at: https://doi.org/10.1109/TPAMI.2018.2858826

Liu, L. and Tan, R.T. (2021). Certainty driven consistency loss on multi-teacher networks for semi-supervised learning. *Pattern Recognition*, vol. 120, p. 108140.
Available at: http://arxiv.org/abs/1901.05657

Longadge, R. and Dongre, S. (2013). Class imbalance problem in data mining review. *CoRR*, vol. abs/1305.1707. 1305.1707.
Available at: http://arxiv.org/abs/1305.1707

Mazurowski, M.A., Habas, P.A., Zurada, J.M., Lo, J.Y. and Tourassi, G.D. (2008). Training neural network classifiers for medical decision making: The effects of imbalanced datasets on classification performance. *Neural networks : The Official Journal of the International Neural Network Society*, vol. 21, no. 2-3, pp. 427–36.
Available at: https://doi.org/10.1016/j.neunet.2007.12.031

Mikolov, T., Sutskever, I., Chen, K., Corrado, G.S. and Dean, J. (2013). Distributed representations of words and phrases and their compositionality. In: *Advances in Neural Information Processing Systems: 27th Annual Conference on Neural Information Processing Systems, Lake Tahoe, Nevada, United States, December 5-8, 2013*, pp. 3111–3119.
Available at: http://arxiv.org/abs/1310.4546

Miotto, R., Li, L. and Kidd, B. (2016). Deep patient: An unsupervised representation to predict the future of patients from the electronic health records. *Scientific Reports*, vol. 6, p. 26094.
Available at: https://www.nature.com/articles/srep26094

Mozafari, A.S., Gomes, H.S., Janny, S. and Gagné, C. (2018). A new loss function for temperature scaling to have better calibrated deep networks. *CoRR*, vol. abs/1810.11586. 1810.11586.
Available at: http://arxiv.org/abs/1810.11586

Müller, R., Kornblith, S. and Hinton, G.E. (2019). When does label smoothing help? In: *Advances in Neural Information Processing Systems: Annual Conference on Neural Information Processing Systems, Vancouver, BC, Canada, December 8-14, 2019*, pp. 4696–4705.
Available at: `http://arxiv.org/abs/1906.02629`

Naeini, M.P., Cooper, G.F. and Hauskrecht, M. (2015). Obtaining well calibrated probabilities using bayesian binning. In: *Proceedings of the 29th Association for the Advancement of Artificial Intelligence Conference on Artificial Intelligence, Austin, Texas, USA, January 25-30, 2015*, pp. 2901–2907. AAAI Press.
Available at: `http://www.aaai.org/ocs/index.php/AAAI/AAAI15/paper/view/9667`

Newman, M. (2005). Power laws, pareto distributions and zipfs law. *Contemporary Physics*, vol. 46, no. 5, pp. 323 – 351.
Available at: `http://dx.doi.org/10.1080/00107510500052444`

Niculescu-Mizil, A. and Caruana, R. (2005). Predicting good probabilities with supervised learning. In: *Machine Learning, Proceedings of the 22nd International Conference, Bonn, Germany, August 7-11, 2005*, vol. 119 of *ACM International Conference Proceeding Series*, pp. 625–632. ACM.
Available at: `https://doi.org/10.1145/1102351.1102430`

Oliver, A., Odena, A., Raffel, C., Cubuk, E.D. and Goodfellow, I.J. (2018). Realistic evaluation of deep semi-supervised learning algorithms. In: *Advances in Neural Information Processing Systems: Annual Conference on Neural Information Processing Systems, Montréal, Canada, December 3-8, 2018*, pp. 3239–3250.
Available at: `http://arxiv.org/abs/1804.09170`

Rigollet, P. (2007). Generalization error bounds in semi-supervised classification under the cluster assumption. *Journal of Machine Learning Research*, vol. 8, no. Jul, pp. 1369–1392.
Available at: `https://hal.archives-ouvertes.fr/hal-00022528`

Sajjadi, M., Javanmardi, M. and Tasdizen, T. (2016). Regularization with stochastic transformations and perturbations for deep semi-supervised learning. In: *Advances in Neural Information Processing Systems: Annual Conference on Neural Information Processing Systems, Barcelona, Spain, December 5-10, 2016*, pp. 1163–1171.
Available at: `http://arxiv.org/abs/1606.04586`

Shorten, C. and Khoshgoftaar, T.M. (2019). A survey on image data augmentation for deep learning. *Journal of Big Data*, vol. 6, no. 1, pp. 1–48.
Available at: `https://journalofbigdata.springeropen.com/articles/10.1186/s40537-019-0197-0`

Simard, P.Y., LeCun, Y., Denker, J.S. and Victorri, B. (2000). Transformation invariance in pattern recognition: Tangent distance and propagation. *International Journal of Imaging Systems and Technology*, vol. 11, no. 3, pp. 181–197.
Available at: `https://doi.org/10.1007/978-3-642-35289-8_17`

Sinharay, S. (2010). Continuous probability distributions. In: *International Encyclopedia of Education (Third Edition)*, pp. 98–102. Elsevier, Oxford.
Available at: `https://www.sciencedirect.com/science/article/pii/B9780080448947017206`

Sohn, K., Berthelot, D., Carlini, N., Zhang, Z., Zhang, H., Raffel, C., Cubuk, E.D., Kurakin, A. and Li, C. (2020). Fixmatch: Simplifying semi-supervised learning with consistency and confidence. In: *Advances in Neural Information Processing Systems: Annual Conference on Neural Information Processing Systems, virtual, December 6-12, 2020*.
Available at: `https://arxiv.org/abs/2001.07685`

Srivastava, N., Hinton, G., Krizhevsky, A., Sutskever, I. and Salakhutdinov, R. (2014). Dropout: A simple way to prevent neural networks from overfitting. *Journal of Machine Learning Research*, vol. 15, pp. 1929–1958.
Available at: `https://dl.acm.org/doi/pdf/10.5555/2627435.2670313`

Su, J., Cheng, Z. and Maji, S. (2021). A realistic evaluation of semi-supervised learning for fine-grained classification. In: *IEEE Conference on Computer Vision and Pattern Recognition, virtual, June 19-25, 2021*, pp. 12966–12975. Computer Vision Foundation / IEEE.
Available at: `https://arxiv.org/abs/2104.00679`

Summers, C. and Dinneen, M.J. (2019). Improved mixed-example data augmentation. In: *IEEE Winter Conference on Applications of Computer Vision, Waikoloa Village, HI, USA, January 7-11, 2019*, pp. 1262–1270. IEEE.
Available at: `http://arxiv.org/abs/1805.11272`

Tang, K., Huang, J. and Zhang, H. (2020). Long-tailed classification by keeping the good and removing the bad momentum causal effect. In: *Advances in Neural Information Processing Systems: Annual Conference on Neural Information Processing Systems, virtual, December 6-12, 2020*.
Available at: `https://arxiv.org/abs/2009.12991`

Thulasidasan, S., Chennupati, G., Bilmes, J.A., Bhattacharya, T. and Michalak, S. (2019). On mixup training: Improved calibration and predictive uncertainty for deep neural networks. In: *Advances in Neural Information Processing Systems: Annual Conference on Neural Information Processing Systems, Vancouver, BC, Canada, December 8-14, 2019*, pp. 13888–13899.
Available at: `https://arxiv.org/pdf/1905.11001.pdf`

Torralba, A., Fergus, R. and Freeman, W. (2008). 80 million tiny images: A large data set for nonparametric object and scene recognition. *IEEE transactions on pattern analysis and machine intelligence*, vol. 30, pp. 1958–70.
Available at: `https://doi.org/10.1109/TPAMI.2008.128`

Verma, V., Lamb, A., Beckham, C., Najafi, A., Mitliagkas, I., Lopez-Paz, D. and Bengio, Y. (2019). Manifold mixup: Better representations by interpolating hidden states. In: *Proceedings of the 36th International Conference on Machine Learning,*

Long Beach, California, USA, June 9-15 2019, vol. 97 of *Proceedings of Machine Learning Research*, pp. 6438–6447. PMLR.
Available at: `https://arxiv.org/abs/1806.05236`

Wood, C., Sullivan, B., Iliff, M., Fink, D. and Kelling, S. (2011). ebird: Engaging birders in science and conservation. *Public Library of Science Biology*, vol. 9, no. 12, pp. 1–5.
Available at: `https://doi.org/10.1371/journal.pbio.1001220`

Wu, T., Huang, Q., Liu, Z., Wang, Y. and Lin, D. (2020). Distribution-balanced loss for multi-label classification in long-tailed datasets. In: *Proceedings of the 16th European Conference at the European Conference on Computer Vision, Glasgow, UK, August 23-28, 2020*, vol. 12349 of *Lecture Notes in Computer Science*, pp. 162–178. Springer.
Available at: `https://arxiv.org/abs/2007.09654`

Xie, Q., Luong, M., Hovy, E.H. and Le, Q.V. (2020). Self-training with noisy student improves imagenet classification. In: *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition, Seattle, WA, USA, June 13-19, 2020*, pp. 10684–10695. Computer Vision Foundation / IEEE.
Available at: `http://arxiv.org/abs/1911.04252`

Yang, Y. and Xu, Z. (2020). Rethinking the value of labels for improving class-imbalanced learning. In: *Advances in Neural Information Processing Systems: Annual Conference on Neural Information Processing Systems 2020, virtual, December 6-12, 2020*.
Available at: `https://arxiv.org/abs/2006.07529`

Yun, S., Han, D., Chun, S., Oh, S.J., Yoo, Y. and Choe, J. (2019). Cutmix: Regularization strategy to train strong classifiers with localizable features. In: *2019 IEEE/CVF International Conference on Computer Vision, ICCV 2019, Seoul, Korea (South), October 27 - November 2, 2019*, pp. 6022–6031. IEEE.
Available at: `http://arxiv.org/abs/1905.04899`

Zhang, H., Cissé, M., Dauphin, Y.N. and Lopez-Paz, D. (2018). Mixup: Beyond empirical risk minimization. In: *6th International Conference on Learning Representations, Vancouver, BC, Canada, April 30 - May 3, 2018*. OpenReview.net.
Available at: `http://arxiv.org/abs/1710.09412`

Zhang, J., Liu, L., Wang, P. and Shen, C. (2019). To balance or not to balance: An embarrassingly simple approach for learning with long-tailed distributions. *CoRR*, vol. abs/1912.04486. `1912.04486`.
Available at: `http://arxiv.org/abs/1912.04486`

Zheng, S., Song, Y., Leung, T. and Goodfellow, I.J. (2016). Improving the robustness of deep neural networks via stability training. In: *IEEE Conference on Computer Vision and Pattern Recognition, Las Vegas, NV, USA, June 27-30, 2016*, pp. 4480–4488. IEEE Computer Society.
Available at: `http://arxiv.org/abs/1604.04326`

Zhong, Z., Cui, J., Liu, S. and Jia, J. (2021). Improving calibration for long-tailed recognition. In: *IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2021, virtual, June 19-25, 2021*, pp. 16489–16498. Computer Vision Foundation / IEEE.
Available at: `https://arxiv.org/abs/2104.00466`

Zhou, B., Cui, Q., Wei, X. and Chen, Z. (2020). BBN: Bilateral-branch network with cumulative learning for long-tailed visual recognition. In: *IEEE Computer Vision Foundation Conference on Computer Vision and Pattern Recognition, Seattle, WA, USA, June 13-19, 2020*, pp. 9716–9725. Computer Vision Foundation / IEEE.
Available at: `http://arxiv.org/abs/1912.02413`