

Mikko Impiö

# **ON IMBALANCED CLASSIFICATION OF BENTHIC MACROINVERTEBRATES: METRICS AND LOSS-FUNCTIONS**

Performance metric and loss-function comparison for  
imbalanced multi-class classification

# ABSTRACT

Mikko Impiö: On imbalanced classification of benthic macroinvertebrates: Metrics and loss-functions

Bachelor of Science Thesis  
Tampere University  
Electrical Engineering  
May 2020

---

Aquatic biomonitoring is an integral part of assessing the state and quality of freshwater systems. An important part of biomonitoring is the identification and classification of benthic macroinvertebrates, a species group containing several indicator species of high interest. Lately, automating the process of identifying these species using visual and chemical systems has gained interest. The methods presented for this often overlook the imbalanced nature of taxonomic data, where the size difference between largest and smallest classes is substantial.

This thesis has two main themes: analyzing the suitability of different performance metrics used to evaluate imbalanced domain classification models, as well as testing methods that could be used to improve the performance of these models. Performance metrics are analyzed from the standpoint of experts with no machine learning expertise, focusing on understandability and visualizations of the metrics. Focus is given on metrics that can be derived from a multi-class confusion matrix, due to the intuitive derivation of these metrics. These metrics are used to produce both single-score and class-wise metrics, that describe the model performance either as whole, or separately for each class. As for classification improvement methods, experiments with different loss functions, rebalancing and augmentation methods are conducted.

This thesis presents as results a comparison of different evaluation metrics with their pros and cons from the biomonitoring point of view. The main argument is that a single metric for describing model performance can be very ambiguous, and if it is possible, further assessment by class-wise metrics should be conducted when comparing models. The results of classification improvement methods did not yield better results than the reference model with the experiments conducted. This thesis also presents a modern reference model trained with a benthic macroinvertebrate benchmark dataset, outperforming most of the current flat classification models in the literature.

Keywords: imbalanced classification, biomonitoring, loss functions, performance metrics, cost-sensitive learning

The originality of this thesis has been checked using the Turnitin OriginalityCheck service.

# TIIVISTELMÄ

Mikko Impiö: Epätasaisen datajakauman ongelmat pohjaeläinten luokittelussa: metriikat ja tavoitefunktio

Kandidaatintyö

Tampereen yliopisto

Sähkötekniikka

Toukokuu 2020

---

Biologinen seuranta on tärkeä osa vedenlaadun seurantaa. Pienilläkin ympäristömuutoksilla voi olla suuria vaikutuksia makeiden vesien lajistoon. Tärkeän osan makeiden vesien ekosysteemeistä muodostaa pohjaeläimet, joiden tunnistamisen automatisointi on herättänyt kiinnostusta viime vuosina uusien koneoppimismenetelmien myötä. Esitetyt menetelmät eivät kuitenkaan ota yleensä huomioon taksonomisen datan suurta epätasaisuutta, jossa suurimman ja pienimmän luokan näytemäärien välillä voi olla monikymmenkertainen ero.

Tässä kandidaatintyössä on kaksi pääteemaa: erilaisten metriikoiden sopivuus pohjaeläinten luokittelumallien arvioinnissa, sekä näiden luokittelumallien parantaminen käyttäen epätasaisuuden huomioon ottavia menetelmiä. Luokittelumallien metriikoissa nostetaan esille menetelmiä, jotka ottavat luokittelijan suorituskyvyn paremmin huomioon myös pienempien luokkien osalta. Työssä esitellään yleisesti käytettyjä ja uusia metriikoita sekä visualisointeja ja tuodaan esille näiden teoreettista taustaa. Metriikat keskittyvät sekaannusmatriisista (confusion matrix) johdettaviin arvoihin, joita käytetään sekä koko mallin suorituskyvyn, että luokkakohtaisten metriikoiden laskemiseksi. Metriikoiden lisäksi tutkitaan menetelmiä, jotka ottavat epätasaisen jakauman huomioon jo luokittelijan luontivaiheessa. Näitä ovat esimerkiksi neuroverkkojen optimoinnissa käytetyt tavoitefunktio, sekä erilaiset koulutusdatajakauman tasapainotus -ja muokkausmenetelmät.

Työn tuloksena esitellään erilaisten metriikoiden hyödyt ja haitat biologisen seurannan näkökulmasta. Pääargumenttina on, että kokonaissuorituskykyä kuvaavan arvon sijaan tulisi tarkastella luokkakohtaisia suorituskykyjä, keskittyen mallin luotettavuuteen läpi luokkien. Luokittelijan parannusmenetelmien tuloksena havaittiin, että epätasaisuuden huomioon ottavat menetelmät eivät juurikaan tuota oletusmallia parempia lopputuloksia. Työssä esitellään myös referenssiluokittelija, joka suoriutuu pohjaeläinten luokittelussa useita kirjallisuudessa esiintyviä yksittäisiä luokittelijoita paremmin.

Avainsanat: epätasainen datajakauma, biomonitorointi, tavoitefunktio, metriikat

Tämän julkaisun alkuperäisyys on tarkastettu Turnitin OriginalityCheck -ohjelmalla.

# CONTENTS

1	Introduction . . . . .	1
2	Related work . . . . .	3
3	Theory . . . . .	5
3.1	Biomonitoring and benthic macroinvertebrates . . . . .	5
3.2	Machine learning . . . . .	9
3.2.1	Statistical learning theory . . . . .	10
3.2.2	Optimization theory . . . . .	11
3.2.3	Neural networks . . . . .	13
3.3	Evaluation metrics for imbalanced classification . . . . .	15
3.3.1	Confusion matrix metrics . . . . .	16
3.3.2	Averaging multi-class classifiers . . . . .	19
3.3.3	F-measure, G-mean . . . . .	20
3.3.4	Bayesian probability . . . . .	22
3.4	Loss functions for imbalanced data . . . . .	23
3.4.1	Cross-entropy loss . . . . .	24
3.4.2	Focal loss . . . . .	25
3.4.3	Class-balanced loss . . . . .	26
3.5	Sampling methods for imbalanced data . . . . .	27
3.5.1	Resampling methods . . . . .	27
3.5.2	Data augmentation . . . . .	28
4	Methods . . . . .	29
4.1	Data . . . . .	29
4.2	Reference model . . . . .	31
5	Comparison of evaluation metrics . . . . .	33
5.1	Traditional metrics for model evaluation . . . . .	33
5.1.1	Cross-validation and averaging . . . . .	34
5.1.2	Confusion matrix and precision-recall -curve . . . . .	36
5.2	Imbalanced multi-class classification metrics and visualizations . . . . .	38
5.3	Qualitative comparison . . . . .	42
6	Comparison of loss-functions and augmentation methods . . . . .	45
6.1	Categorical cross-entropy . . . . .	46
6.2	Focal loss . . . . .	46
6.3	Class-balanced loss . . . . .	48
6.4	Resampling and data augmentation methods . . . . .	49
6.4.1	Under- and oversampling . . . . .	49
6.4.2	Data augmentation . . . . .	51

6.5 Results . . . . .	52
7 Conclusion . . . . .	54
References . . . . .	56
Appendix A Appendix . . . . .	62

## LIST OF FIGURES

3.1	Relationship between the abundance and the amount of species . . . . .	7
3.2	Different kinds of benthic macroinvertebrates . . . . .	9
3.3	Cross-entropy and focal loss as a function of probability . . . . .	25
4.1	Taxonomic resolution of the image data . . . . .	30
4.2	Class sizes of the full dataset . . . . .	31
5.1	Reference model test split 1 confusion matrix . . . . .	37
5.2	Reference model split 1 precision-recall curve . . . . .	38
5.3	Reference model confusion matrix metrics and F1 score . . . . .	39
5.4	Reference model positive performance metrics, G-mean and F1 score . . .	40
5.5	Reference model cumulative positive performance metrics and F1 score . .	41
5.6	Example of two different classifiers with the same macro-averaged F1 score, but drastically different overall performance . . . . .	41
6.1	Weighted cross-entropy comparisons for normalized and inverse frequency weights . . . . .	47
6.2	Focal loss ( $\alpha = 1, \gamma = 2$ ) performance comparison . . . . .	48
6.3	Class-balanced loss performance comparison . . . . .	49
6.4	Reference model continued with oversampled data performance . . . . .	51
6.5	Performance of a model trained with augmented data . . . . .	52
A.1	Focal loss F1 scores plotted against reference model F1 scores in ascend- ing order . . . . .	65
A.2	Focal loss ( $\alpha = 1$ ) cumulative plots against reference model . . . . .	66
A.3	Focal loss F1 improvements against reference . . . . .	67

## LIST OF TABLES

3.1	Comparison of biosurvey organism assemblages . . . . .	8
3.2	Confusion matrix for binary classification . . . . .	17
3.3	Multi-class confusion matrix . . . . .	17
3.4	Multi-class confusion matrix as binary one-vs-rest matrix with class <i>A</i> as the class under inspection . . . . .	18
3.5	Common classification evaluation metrics . . . . .	19
4.1	Taxa labels . . . . .	30
5.1	Error rate and LCSE metrics for the 4-fold cross-validated reference clas- sifier with comparison to classifier trained with same data by Årje et al. . . . .	34
5.2	Micro- and macro-averaged performance of the reference classifier for each cross-validation split . . . . .	34
5.3	The different multi-class cross-validation averaging combinations for F1 score	35
5.4	Comparison of different metrics . . . . .	44
6.1	Performance of models trained using resampled datasets . . . . .	50
6.2	Performance comparison of different models using single-score metrics . . . . .	53
A.1	Taxa with their taxonomic classifications and class sizes . . . . .	62
A.2	Reference model confusion matrix values summed over folds (micro-averaging)	63
A.3	Reference model confusion matrix values . . . . .	64

## LIST OF SYMBOLS AND ABBREVIATIONS

acc	Accuracy
ANN	Artificial neural network
CB	Class-balanced loss
CE	Cross-entropy
CNN	Convolutional neural network
err	Error rate
F1	F1 score, harmonic mean of precision and recall
FL	Focal loss
FN	False negative
FP	False positive
FPR	False positive rate, 1-TNR
ISO	International Organization for Standardization
LCSE	Level-aware context-sensitive error
MCC	Matthews correlation coefficient
MLP	Multilayer perceptron
NPV	Negative predictive value
PPV	Positive predictive value (Precision)
RGB	Red green and blue
TN	True negative
TNR	True negative rate (Specificity)
TP	True positive
TPR	True positive rate (Sensitivity, Recall)

# 1 INTRODUCTION

Aquatic biomonitoring is an integral part of assessing the state and quality of waterbodies. This monitoring of freshwater environments has become increasingly important in the past decades, as the growing human population increases the demand for fresh water [1]. Even United Nations states as one of their Sustainable Development Goals to "*Ensure availability and sustainable management of water and sanitation for all*" [2]. The water crisis increases the demand for reliable ways of determining the quality of existing freshwater systems, and biomonitoring provides a method for this.

Benthic macroinvertebrates, small invertebrates that inhabit freshwater systems, are an excellent species group for biomonitoring due to their diversity and environmental requirements [3]. Identification of these species is hard and has been done to date by experts with knowledge on the taxonomic differences between these species. Lately, it has been shown that modern computer vision systems can achieve accuracy close to these human experts. [4] Automated image analysis systems have been regarded as tools that can possibly save a lot of scientists' time, and have thus received more attention in the recent years [4, 5, 6]. However, these systems, usually trained with large amounts of image data, have some drawbacks stemming from imbalanced, long-tailed datasets. The number of training images for different taxonomic classes can vary a lot, which can result in incorrect or unreliable classification of rare species. Unfortunately, these rare species can also be the ones biologists are most interested in, producing an important problem to consider. There exists a need to make these identification systems more reliable, ensuring the system also identifies classes with less available training data.

The primary users of automatic biomonitoring systems are mostly biologists and environmental scientists, who often are not machine learning experts, and are hesitant of the reliability of the new technology [4]. In addition to having sufficient performance, automatic systems need to be understandable, their decisions should be somewhat explainable, and the systems should be easy to evaluate. The strengths and weaknesses of an automatic system should be clear for the user if the system is used alongside human processes. This calls for clear evaluation metrics and visualizations of system performance. Increased transparency between a black-box model and a human can hopefully strengthen the confidence in using the systems, and possibly save the experts' time in the long run.

Often the performance evaluation of automated biomonitoring systems relies on fairly simple metrics such as accuracy or F1-score[4, 6, 7]. These metrics are often chosen

probably due to their commonness and ease of use, and the reasons of the choice are rarely argued in-depth in the studies. The choice of an evaluation metric is not a easy task for imbalanced data, as the best metric can change according to the use case, data distribution, or ease of use, for example. However, the comparisons of metrics themselves are rarely discussed in biological monitoring literature, even though it is a critical part of comparing different methods. One of the goals of this thesis is to compare and analyze metrics that are commonly used, deemed most suitable for automatic biomonitoring, and easy to intuitively understand, in a one place. This hopefully should help the non-machine-learning experts in the challenging task of choosing the most suitable metrics for automated biomonitoring problems.

This thesis focuses on evaluating different performance metrics for multi-class classifiers trained with imbalanced data, as well as testing different classifier-improvement techniques using methods that take the imbalance to account. The work presents a reference classifier trained with a benchmark dataset for benthic macroinvertebrate classification achieving initially an 92.50% accuracy on the test set. The dataset used for all the experiments was collected by the Finnish Environment Institute, and is presented in Raitoharju et al. [8]. One of the arguments of this thesis is, however, that the commonly used accuracy metric can be a misleading metric especially for imbalanced data, and that the choice of the performance metric should be chosen with care. Although the reference classifier achieves a "good" score of over 90% accuracy, the classifications for a fifth of the 39 classes are unreliable, which is demonstrated by different metrics and visualizations. Attempts at improving the shortcomings of the reference classifier are then presented, in form of alternative loss functions during neural network optimization, as well as in terms of resampling and data augmentation.

Chapter 2 explores the related literature on automated biomonitoring as well as studies on imbalanced classification, performance metrics and cost-sensitive learning. The theory Chapter 3 presents background in benthos identification, necessary theory on machine learning, background on the performance metrics used in this thesis, and the necessary optimization theory background to understand the importance of choosing loss-functions in optimization-based representation learning. The optimization Section 3.4 also presents the loss-functions used in the experiments and the necessary theory behind understanding them.

The methods Chapter 4 presents the dataset used for training the reference classifier, as well as the details on the reference classifier and the training process itself. The main findings of the thesis are presented in Chapters 5 and 6: the suitability of different performance metrics for evaluation of biomonitoring models, and the methods for improving them using imbalance-sensitive loss-functions.

Finally, Chapter 7 presents the conclusions of the thesis, with suggestions on the most suitable performance metrics and visualizations, as well as discussion on the classification improvement methods. Considerations on further studies on the topic are also made here.

## 2 RELATED WORK

Autonomic species identification has been well researched in the past years, and Kalafi, Town and Dhillon [5] provide a good overview to several studies in image based classification. Taxonomic classification is an classification task at its purest form, and popular classification methods, such as k-nearest-neighbors (KNN), support vector machine (SVM), and artificial neural networks (ANN), including multilayer perceptron (MLP) and convolutional neural network (CNN), approaches have been proposed for the task [5]. Some of the earliest examples of taxonomical classification are the DiCANN system developed for classification of dinoflagellate by artificial neural networks in 1996 by Culverhouse et al. [9], and DAISY, a vision-based system for identifying invertebrates was presented by O'Neill et al. [10] in 2000.

The field of machine learning and computer vision has achieved considerable progress in the past decade. Neural network based representation learning has proved to be one of the best methods for classification tasks, given one has enough quality data. SVM and MLP based approaches in macroinvertebrate classification have been proposed by Kiranyaz et al. [7] and Joutsijoki et al. [6]. These methods rely on feature-engineering where hand-crafted features calculated from images are fed to an neural network or SVM classifier. More recent studies have shown, that if there is enough available training data, feature extraction using convolutional neural networks can yield better results than traditional feature-engineering methods. Examples of using CNNs in taxonomical classification include Raitoharju et al. [8] using CNNs for benthic macroinvertebrate classification, and Wei et al. [11] using a more sophisticated CNN-based method for recognizing bird parts and using them to classify bird species.

Although modern machine learning methods are data-intensive, there has been a lack of publicly available benchmark datasets for macroinvertebrate classification [5]. A benchmark dataset by Raitoharju et al. [8] provides a good reference point for comparing automatic taxa classification methods. The iNaturalist dataset [12] is a recent, highly imbalanced, taxonomic classification dataset. The dataset has been used in more general learning problems as a "difficult dataset" [13], as well as in imbalanced classification problems due to its characteristic unbalanced distribution [14].

Most of the above mentioned studies use accuracy or classification error (1-accuracy) as their main evaluation metric. Many studies focus on novel classification methods, but little attention has been given to different metrics that could be used for more reliable evaluation. In other fields evaluation metrics, their differences, and advantages in different

applications have been well studied [15, 16, 17, 18, 19]. A review by Powers [18] provides an extensive overview on different confusion matrix metrics, their advantages, and disadvantages. A similar analysis of several performance metrics used in machine learning applications was performed by Sokolova and Lapalme [19]. Deeper insight on the F1-score and its probabilistic interpretation was presented by Goutte and Gaussier [15]. A study by Bekkar, Djemaa and Alitouche [16] focuses on suitable evaluation metrics used on imbalanced data.

Taxonomical data is often highly imbalanced and long-tailed, meaning that the majority of samples or species are in the largest few classes, with the rest of the samples or species being distributed among a large number of classes [12]. Even though the class imbalance problem is fairly well-studied in its general form [20, 21, 22, 23], it seems that often in many other studies imbalanced domains are treated similarly as balanced domains. This importance and effect of the domain set on classification is highlighted in a study by Japkowicz and Stephen [20]. Also, Krawczyk [21] discusses the various methods of assessing an imbalanced classification problem, from domain modification by resampling to cost-sensitive learning.

Generally, the methods of assessing imbalanced data can be split into two areas: domain modification and cost-sensitive learning. Domain modification refers to methods that artificially alter the domain set to produce a more balanced dataset, or by producing artificial samples by augmenting the data [24, 25, 26, 27]. Wang and Perez [24] explore the effectiveness of different augmentation methods commonly used in deep learning. Under- and oversampling are common methods used in rebalancing, but it has been argued by Drummond, Holte et al. [28] that undersampling is more effective. A very common method of oversampling is to add synthetically added samples based on existing ones, presented by Chawla et al. [29].

Cost-sensitive methods alter the training process by giving more attention to certain samples, which can be chosen by, for example, size or relative importance [14, 30, 31, 32]. The field of cost-sensitive learning is broad and includes also interpretations like the cost matrix and cost-sensitive decision making [31]. In this thesis, when cost-sensitive learning is discussed, the main focus is on optimization functions used while training neural networks. Several improvements in these methods have been proposed recently, like the focal loss [32] and class-balanced loss [14] discussed later.

## 3 THEORY

This chapter presents the necessary background theory behind the findings and experiments presented later in Chapters 5 and 6. Most of the concepts are explained from the ground up to provide theoretical background also for non-machine-learning experts (biologists, environmental scientists) that might be interested in the findings.

First, Section 3.1 discusses the background and motivation for biomonitoring in general, highlighting the importance of benthic macroinvertebrates as a species group. The motivation behind the interest in automating the identification process is also brought up.

Section 3.2 goes briefly through essential concepts in machine learning, learning theory, and optimization theory that are referenced later in the thesis. The section also introduces some notation conventions that the later theory uses. Source material notation is often changed to a format consistent with other material in this thesis.

Section 3.3 presents the theory behind most common evaluation and performance metrics used in machine learning. The confusion matrix and metrics derived from it are discussed, with a focus on metrics commonly used with imbalanced data.

Sections 3.4 and 3.5 discuss the two main methods for assessing problems with imbalanced data: cost-sensitive learning and resampling. Cost-sensitive learning, assigning different weights to different classes during training, is discussed in form of loss functions in Section 3.4, and Section 3.5 goes briefly through the most common resampling and data augmentation methods that are used later in Chapter 4.

### 3.1 Biomonitoring and benthic macroinvertebrates

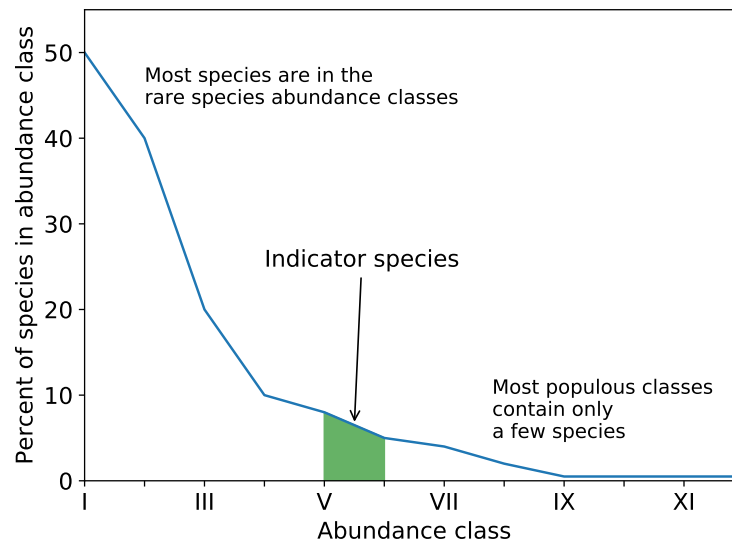
Biomonitoring, or aquatic biomonitoring, refers to the practice of assessing the ecological state of an aquatic environment using a variety of methods [33, 34]. Using aquatic biomonitoring for water quality assessment has scientific roots in 19th century London, where river species communities and organic pollution were linked together after a severe cholera epidemic [35, 36]. Biomonitoring attempts to track and study the environment's response to external disturbances and to improve understanding of the chemical, biological, and physical components of the habitat [37]. Identification of local organisms is an essential part of the process of assessing the condition of waterbodies and their ecosystems. Since the condition of a waterbody is affected by the surrounding environment, biomonitoring of these aquatic environments is crucial in monitoring ecosystems

as a whole, making it possible to detect pollution, ecological degradation, and possible effects of climate change over time [38]. Freshwater monitoring focuses on monitoring freshwater ecosystems, such as rivers, streams, lakes, and ponds, where the effect of the surrounding environment and possible pollution can be significant. Since fresh water is an increasingly valuable resource integral to all ecological and social activities, monitoring the quality of the existing sources of fresh water has become an pressing issue worldwide [1, 39].

Freshwater quality can be assessed both biologically and chemically [3]. Biological assessment and monitoring are based on the idea that the presence or absence of certain species in different areas can give valuable information about the ecological environment as a whole [33]. In the case of freshwater habitats, the presence or absence of different species of fish, algae, or insects, for example, can give scientists valuable information on the condition of a waterbody. These key species are usually referred to as *indicator species*. To be specific, the term *indicator species* can refer to different concepts in biology and ecology, for example, referring to the dominant species in an area, or a primary node in the ecological relationships of species that has a significant impact on other species if the species is lost [40]. However, in most contexts, including this thesis, the term refers to a species or an assemblage of species that have strict requirements on different characteristics of their habitat, thus indicating possible abnormalities in the health of the ecosystem [41].

When an indicator species is present, the environment fulfills the strict habitat requirements of the species. For example, some moss and plant species have strict requirements on the air quality of their habitat and thus can only be found far from human populations and air pollution [42]. Choosing these indicator species or assemblages of species is an crucial part of ecological assessment. For well-known indicators, such as pollution, choice of species is relatively straightforward and well-researched, but in more specific situations lacking historical evidence the choice should be made carefully [41, 43]. In general, the indicator species or assemblage of species should be abundant in the area for ease of sampling [41], well-researched [44] and taxonomically consistent [41] in addition to displaying strict habitat requirements specified above. Because of these habitat requirements, indicator species are generally rather scarce, but abundant enough to make reliable assessments of the habitat. This relationship is illustrated in figure 3.1 where species are divided into abundance classes, with higher class number meaning a more abundant and populous species. The figure 3.1 shows a "long tail" effect, where the numerical amount of rare species is large, with a few abundant species dominating the habitat. Well-suited indicator species can be found in the middle, with a diversity of different species with special environmental requirements, but enough abundance for surveying.

Biological survey techniques usually focus on assessing an assemblage of species in the habitat under survey. These groups can reflect the overall ecological quality of the habitat and can display the effects of stressors working on the environment. Three main assem-



**Figure 3.1.** Relationship between abundance (exponential scale) and amount of species, with indicator species in the abundance classes V and VI. Modified from [41].

blages used in biosurveys can be identified: periphyton, benthic macroinvertebrates, and fish. Some characteristics of the three communities are presented in table 3.1. [3]

Periphyton refers to a mixture of algae, bacteria, and other biological matter that can be found from aquatic environments. The main advantages of using periphyton include the ease of sampling and well-defined methods for evaluating the characteristics of algal communities [3, 45]. For most people, fish are the most visible organism in freshwater environments. The importance of fish for humans gives a special meaning in their monitoring. Due to this history, the characteristics of fish are well-known. Fish also account for almost half of the endangered vertebrate species in the United States [46].

### **Benthic macroinvertebrates**

From the three main assemblage classes, benthic macroinvertebrates are the most commonly used indicator group for aquatic biomonitoring. Their abundance, broad range of species and taxons, limited mobility, and characteristic responses to changes in external factors make them an ideal biological indicator [3, 33]. The term *benthic macroinvertebrate* refers to a group of species inhabited in freshwater ecosystems. Figure 3.2 illustrates the large variety of species considered benthic macroinvertebrates. Generally, all freshwater invertebrates that can be seen with the naked eye belong to this group. Most species are insects, but other invertebrates such as snails, worms, and clams also satisfy the definition. Due to vast diversity of species with unclear boundaries, benthos are usually classified and grouped on different taxa levels, usually species, genus, or family level [3]. One location in a waterbody can contain tens of different taxa [47], making statistical comparison methods of taxa assemblages between locations and across time possible. Since different taxa of benthos react differently to human influences in forms of pollution

**Table 3.1.** Comparison of biosurvey organism assemblages [3, 45]

	<b>Periphyton</b>	<b>Benthic macroinvertebrates</b>	<b>Fish</b>
Life cycle	Rapid for algae, differs for other organisms	Approximately one-year	Several years
External impact effect	Short-term	Short-term with possibility to track long-term effects	Long-term
Food chain role	Serves as an important food source for other organisms	Serves as a primary food source for fish	Top of the food chain, accumulating pollutant effects from lower levels
Sampling	Easy	Relatively easy and has minimal effect on other organisms	Easy
Pollutant sensitivity	Algae and other biomass can be sensitive to pollutants even with low concentrations	Diverse resilience to pollutants between taxa provides strong information for evaluating cumulative effects	Differs between species but is usually well-known
Abundance	Abundant in most environments	Abundant in most environments	Depends on the environment
Mobility	Low	Limited	High

or chemicals [33], analyzing large assemblages can yield good and reliable statistics of the ecological environment. These assemblage samples are collected from the field with a variety of methods to be further analyzed in the laboratory.

Most of the laboratory processing for benthic macroinvertebrates is done by hand. The methods and protocols for this are well-defined [3], but the process is still very time-consuming and needs a taxonomic expert to perform [4, 5]. Several challenges have been identified in species identification: there is a lack of an agreed list of described species, species' taxonomical classifications are unclear, specialists are scarce, and their education is time-consuming and difficult [48]. Because of these reasons, automatic taxa recognition methods have been proposed, and some of them are presented more in-depth in Chapter 2. However, while some of these methods have shown promising signs in automating taxa recognition, biologists are often skeptical of their performance since



**Figure 3.2.** Different kinds of benthic macroinvertebrates sampled from the benthic macroinvertebrate benchmark dataset [8]

the expertise needed for taxa recognition is high, and some biologists might be afraid of the possible new methods gaining a human-level proficiency in a challenging field [4]. However, if taxa recognition could be done accurately, biologists could use this saved time and their expertise for other vital tasks in biological monitoring.

### 3.2 Machine learning

The world as we know it contains a vast amount of structure and information for what intelligent systems, like human beings, can assign different signs and meanings to. The problem of finding information patterns from data is known as *pattern recognition*, a task human beings are very capable of doing intuitively [49]. Humans recognize patterns all the time on different abstraction levels: visual signals can be interpreted as different objects, and low-level concepts can be combined to produce higher-level patterns and concepts. Intelligent systems are capable of producing *generalizations* from data presented to them, making them able to recognize patterns and objects also from previously unrepresented data. The ability of artificially intelligent systems to learn to recognize patterns from raw data and generalize from them is known as *machine learning* [50].

The field of machine learning has progressed significantly during the 2010s, and has gained a lot of interest from both academic and industrial sectors. Especially *computer*

*vision* has gained significant progress, where advanced pattern recognition and machine learning techniques have improved the big problems of classification and object recognition. The problem present in this thesis, automatic biomonitoring by classification of benthic macroinvertebrates, falls into the classification subfield inside computer vision.

This section focuses on three important areas in machine learning: statistical learning theory, optimization and representation learning by neural networks. These areas are important considering the problems presented in this thesis: statistical learning theory provides background for the notation and methods used in both performance metrics and classification improvement methods, and optimization theory introduces the necessary theoretical background for understanding the loss-functions discussed later. Finally, the neural network section introduces concepts in representation learning necessary to understand the deep learning methods used later.

### 3.2.1 Statistical learning theory

The goal of *supervised learning* is to build a function  $f(\mathbf{x})$  that maps inputs  $\mathbf{x}$  from a domain set  $\mathbf{x} \in \mathcal{X}$  to outputs  $y$  in a label set  $y \in \mathcal{Y}$ , using known input-output pairs  $(\mathbf{x}, y)$  as training data. Training data is a set  $S = \{(\mathbf{x}_1, y_1), (\mathbf{x}_2, y_2), \dots, (\mathbf{x}_n, y_n)\} \subseteq \mathcal{X} \times \mathcal{Y}$ , where each *instance* of a domain point  $\mathbf{x}_i$  can refer to, for example, an image of a benthic macroinvertebrate and the *label*  $y_i$  to the invertebrate's taxonomic class. The training samples (image-label pairs) are used to build a *classifier*, or *model*  $f(\mathbf{x})$  that attempts to replicate the original, unknown mapping  $h(\mathbf{x})$  as well as possible. The training of the classifier  $f(\mathbf{x})$  is done in a way that should minimize the difference between the classifier output  $\hat{y} = f(\mathbf{x}; \theta)$  and the true label values  $y$ . [51, 52] Here the notation  $f(\mathbf{x}; \theta)$  refers to a model  $f(\mathbf{x})$  built with parameters  $\theta$ . The method of calculating the difference depends on the data that is used and can be referred to as a performance metric  $P$  [50, p. 272]. In the benthic macroinvertebrate case, the target set is *categorical*, finite set  $\mathcal{Y} = \{1, 2, \dots, C\}$ , where  $C$  is the amount of classes.

The domain space  $\mathcal{X}$ , for example all possible images of benthic macroinvertebrates, follows a probability distribution  $p_{data}$  called the *data-generating distribution*. Because it is generally impossible that all possible samples from the domain set are gathered for training (no need for learning a model since we have knowledge of the full set), the collected training dataset  $S$  follows an empirical distribution  $\hat{p}_{data}$ . The purpose of statistical learning is to find a function  $f(\mathbf{x})$ , using data sampled from  $\hat{p}_{data}$ , that is able to classify samples from the underlying distribution  $p_{data}$  correctly. [50, p. 109, 272]

It is important to understand the effect the dataset and its distribution  $\hat{p}_{data}$  has on training. In the benthic macroinvertebrate case, all possible RGB images with a height of  $m$  pixels, a width of  $n$  pixels, and a depth of three color channels that *represent a benthic macroinvertebrate* form the domain set  $\mathcal{X}$ . Note that this is a subset of all possible RGB images that can represent anything from coffee cups to sunsets. If this superset of all possible images is  $\mathcal{X}_{all}$ , then  $\mathcal{X} \subset \mathcal{X}_{all}$ . When our training algorithm produces a function

$f : \mathcal{X} \rightarrow \mathcal{Y}$ , interesting problems arise when the classifier is given an image from  $\mathcal{X}_{all} \setminus \mathcal{X}$ , i.e., an image that *is not* a benthic macroinvertebrate, or is at least a lot different than a "normal" image of this type. This image, even though it does not represent a benthic macroinvertebrate, is still a valid input for the classifier and is mapped to a label output in  $\mathcal{Y}$ . The problem above is highlighted when the data-generating and empirical distributions differ a lot. Say a model  $f'(\mathbf{x})$  is trained with a dataset  $S'$  (for example augmented data) sampled from a different domain set  $\mathcal{X}' \subset \mathcal{X}_{all}$ , producing the same nominal outputs  $\mathcal{Y}$  as  $f(\mathbf{x})$ . The model can easily be mistaken to behave the same as  $f(\mathbf{x})$  since the output space is the same; however, because the domain set distributions are different, the model *is different* and can behave unexpectedly.

Usually a model is trained with a goal of minimizing some performance metric  $P$  between the predictions of the model  $\hat{y} = f(\mathbf{x}; \theta)$  and the true values  $y = h(x)$ . If this performance metric is used to build the model  $f(\mathbf{x}; \theta)$  and find its parameters  $\theta$ , then the performance metric  $P$  is the same as the objective function value  $J(\theta)$  and the objective of minimizing (or maximizing) the performance metric is known during training. In this case, the problem can be formulated as a regular optimization problem, where the goal is to find parameters  $\theta$  that minimize the objective function  $J(\theta)$ . [50, p. 271]. Most optimization algorithms rely on the differentiability of objective functions in order to calculate the direction of optimization steps. Often in a machine learning setting, the performance metric  $P$  is rarely differentiable and thus a objective function  $J(\theta)$  different from  $P$  must be used during training. Poor choice of an objective function can lead to poor performance or a model that is hard to train, and ultimately the performance metric determines how the model results are interpreted.

### 3.2.2 Optimization theory

The general form of an optimization problem is

$$\begin{aligned} & \text{minimize } f_0(\mathbf{x}) \\ & \text{subject to } f_i(\mathbf{x}) \leq b_i, i = 1, \dots, m \end{aligned} \tag{3.1}$$

where the vector  $\mathbf{x}$  is the *optimization variable* we wish to find in order to optimize the *objective function*  $f_0(\mathbf{x})$  [53]. If the problem is a minimization problem the objective function can be called as a *cost function* or *loss function* [50]. These terms can be used interchangeably in traditional optimization, but in learning algorithms they have special meanings that are discussed later. The vector  $\mathbf{x} = (x_1, \dots, x_n)$  usually is a  $n$ -dimensional vector and the objective function maps values from  $\mathbb{R}^n$  to  $\mathbb{R}$ . The functions  $f_i(\mathbf{x}), i = 1, \dots, m$  are called the *constraint functions* of the problem, and can be split up to functions

$$\begin{aligned} h_i(\mathbf{x}) &= 0, i = 1, \dots, r \\ g_j(\mathbf{x}) &\leq 0, j = 1, \dots, s \end{aligned} \tag{3.2}$$

The *feasible set*  $\Omega \subset \mathbb{R}^n$  constrained by the constraint functions is now

$$\Omega = \{\mathbf{x} \in \mathbb{R}^n | h_i(\mathbf{x}) = 0, \text{ for } i = 1, \dots, r \text{ and } g_j(\mathbf{x}) \leq 0, \text{ for } j = 1, \dots, s\}. \quad (3.3)$$

The values  $\mathbf{x} \in \Omega$  are said to be feasible solutions. If  $\Omega = \mathbb{R}^n$ , the problem is said to be *unconstrained*, which is usually the case with learning algorithms. The value  $\mathbf{x}^*$  that minimizes (or maximizes) the objective function is called the *optimum* or the *solution* to the optimization problem. [53, 54]

Usually optimization problems related to training machine learning algorithms involve a minimization problem, where the objective function can be split to two different functions. The final objective function in learning is called the *cost function* and marked as  $J(\theta)$  where  $\theta$  is the parameter vector. This cost function usually is the average of *loss function*

$$L(f(\mathbf{x}; \theta), y) \quad (3.4)$$

values over the training set. Here  $f(\mathbf{x}; \theta)$  is the output of the model using parameters  $\theta$  with input  $\mathbf{x}$ , and  $y$  is the true output of the data  $\mathbf{x}$ . [50] The trained model or classifier attempts to replicate an unknown mapping  $h(\mathbf{x}) = y$  from inputs  $\mathbf{x}$  to outputs  $y$  as well as possible. The theoretical background behind learning models is discussed in the beginning of Section 3.3. The cost function  $J(\theta)$  gets now the value

$$J(\theta) = E[L(f(\mathbf{x}; \theta), y)], \quad (3.5)$$

where  $E$  is the expectation function and training data pairs  $(\mathbf{x}, y)$  are sampled from an empirical distribution  $\hat{p}_{data}$ . This empirical distribution should be similar to the true *data-generating distribution*  $p_{data}$ , from where the training data is sampled from. To create this distribution, all possible points from a domain set  $\mathcal{X}$  with the corresponding labels from  $\mathcal{Y}$  would be needed. This would form an optimization problem

$$\min_{\theta} J^*(\theta) \quad (3.6)$$

where  $J^*(\theta) = E[L(f(\mathbf{x}; \theta), y)]$  but training data pairs would be sampled from  $p_{data}$  in contrast to equation 3.5. Since this is not possible in most cases, the optimization problem reduces to optimizing the function 3.5 with respect to  $\theta$ . The cost associated with equation 3.6 is known as *risk*, and the cost minimized using equation 3.5 is known as *empirical risk*. [50]

## Convexity

A function  $f : K \rightarrow \mathbb{R}$  is convex if for all  $x, y \in K$  and  $\lambda \in [0, 1]$

$$f(\lambda x + (1 - \lambda)y) \leq \lambda f(x) + (1 - \lambda)f(y). \quad (3.7)$$

If a function is not convex, it is concave. Convex functions have a special property in terms of optimization, since the solution  $\mathbf{x}^*$  is a *global solution/minimum*, meaning that for all  $\mathbf{x}$ ,  $f(\mathbf{x}^*) \leq f(\mathbf{x})$ . If the function is concave, it is possible to find a global minimum, but this is not guaranteed. A optimization algorithm can converge into a *local minimum*, where the smallest value of  $f(\mathbf{x})$  is achieved in a local neighborhood  $B$  where  $f(\mathbf{x}^*) \leq f(\mathbf{x})$  for all  $\mathbf{x} \in B$ . The neighborhood  $B$  is defined as open set  $B(\mathbf{x}^*, r) = \{\mathbf{x} \in \mathbb{R}^n \mid \|\mathbf{x} - \mathbf{x}^*\| < r\}$ . In other words, the local minimum is a point that is smaller than all the other values in its immediate vicinity. [53, 54]

Most real-life optimization problems and loss-functions are not convex. This can lead to problems where the optimization process converges to local minimum, thus not finding the best solution in a global minimum. It is also important to understand that because of this a machine learning algorithm that finds its model by optimization is not necessarily the best possible. Especially with distribution problems discussed before, and if the loss-function is chosen poorly, the local optimum might be one that, for example, maximizes only the accuracy of the largest classes. Because of this the choice of loss-function is important especially in an unbalanced domain.

### 3.2.3 Neural networks

A subfield of machine learning where classification problems often fall into, is called *representation learning*. In representation learning, the systems learns not only the mapping between an input and an output, but also the *representation* of the data itself. Classical machine learning methods rely on hand-crafted features in order to produce a representation of the data, for example different statistical values can be extracted from an image and these can be used as an input for the machine learning system. Representation learning attempts to learn the most suitable form of representation straight from the data, without human interaction. [50]

*Deep learning*, a subfield of representation learning, produces new levels of representation abstractions from lower-level representations by adding several layers of functions to the model. Most modern classification models are based on deep representation learning, for example convolutional neural networks discussed in this thesis. CNNs contain several layers of *convolutional* and *pooling* operations, which make it possible to learn different levels of features from an grid-like structure, such as an image. [50] In essence, earlier layers of CNNs learn low-level feature representations such as edges and circles, and lower levers combine these to produce more abstract features, such as arms and legs. The final abstract features are then used as an input to *fully connected layers* containing nonlinear functions capable of learning the mapping from a feature set to an output.

This section goes through the necessary theory behind deep learning to provide background for the later chapters. Focus is given on interpreting outputs of neural networks as probability density functions, as well as transfer learning and techniques for this. The

former relates closely to the loss-functions used later, the latter to the problem of taxa recognition and classification in general.

## Mapping outputs

When our model maps inputs to a discrete unordered set  $\mathcal{Y}$ , the target set is said to be *categorical*, and a model that outputs categorical variables is called a *classifier* [55]. When classifying benthos to 39 different taxa classes, the target set is  $\mathcal{Y} = \{1, 2, \dots, 39\}$ , with respecting cardinality of  $C = 39$ . The numbers in this set are each representing a single taxa class; for example, the number 1 refers to the taxa class *Agapetus*. These values are unordered and mutually exclusive, meaning that one image belongs in a single class. A classification problem that maps an input to a single categorical output is referred to as *multiclass classification*, in contrast to *binary classification*, where target set cardinality is 2, and *multilabel classification*, where one image can map to a combination of categorical values [56].

Often the output of the model  $f : \mathcal{X} \rightarrow \mathcal{Y}$  is not categorically determinable, meaning that the model does not output a single class prediction  $\hat{y}$ , but instead a real valued probability

$$q(y = c|\mathbf{x}) \in [0, 1] \quad (3.8)$$

for each  $c \in \mathcal{Y}$  and input vector  $\mathbf{x}$  [57]. Now instead of a single value  $\hat{y} \in \mathcal{Y}$ , we have a  $C$ -dimensional vector

$$q(y|\mathbf{x}) = \{q(y = 1|\mathbf{x}), \dots, q(y = C|\mathbf{x})\} \quad (3.9)$$

satisfying

$$\sum_{c=1}^C q(y = c|\mathbf{x}) = 1 \quad (3.10)$$

which is analogous to the probability mass function of a random variable. Often the probability vector output is the result of an *softmax function* as the final layer of a deep learning model, normalizing values ranging between  $(-\infty, \infty)$  to values in range  $(0, 1)$ . The softmax function is defined by

$$\text{softmax}(\mathbf{z})_i = \frac{e^{z_i}}{\sum_j e^{z_j}}, \quad (3.11)$$

where  $\mathbf{z}$  is an input vector of real values. The value of  $\text{softmax}(\mathbf{z})_i$  corresponds to the value  $q(y = c_i|\mathbf{x})$ , if  $c_i$  is the  $i$ th class. Softmax function makes it desirable for the model to learn weight parameters that produce as large as possible values for the final  $z_i$  value corresponding to the correct output label, since this produces a softmax output close to 1 for this class and close to 0 for others. As it can be later seen from the cross-entropy loss-function, this is a desired output for the correct class prediction vector. [50]

Using a probability distribution for each classification instead of a nominal value opens

up methods from information theory for assessing the performance of a classifier in a differentiable way. When replacing a hard-to-optimize function (such as 0-1 -loss from equation 3.13) with a similarly behaving function that is easier or more efficient to optimize (functions derived from a probability distribution) , the function is called a *surrogate loss function*. [50, 56]

Reducing a distribution  $p(y|\mathbf{x})$  into a categorical value  $\hat{y}$  for final evaluation can be done in different ways. One of the most common methods for this is finding the *maximum a posteriori* value, or the most likely label from

$$\hat{y} = f(\mathbf{x}) = \underset{c}{\operatorname{argmax}} p(y = c|\mathbf{x}) \quad (3.12)$$

which is the estimate for the most probable class.

Mapping outputs this way is not unique to neural networks or deep learning, but is arguably the most common way in classification problems involving neural networks.

## Transfer learning

*Transfer learning* refers to the learning model adapting to a new task or mapping to be learned, using knowledge gained from a previously learned task. The idea behind this is that low-level representations learned in the first task are common to the second task, and the learning system needs only to learn the higher level representations and the mapping. In CNNs, the low-level features are often same for several domains. A model that learns from a large enough dataset can utilize these learned low-level features to learn a more specialized task faster. [50]

There are generally seen three methods for using a pretrained CNN in a new domain: full network training, freezing the earliest convolutional layers in a network, and freezing all convolutional layers in a network. By fixing the convolutional layer weights to be untrainable (freezing the layers), the layers can now be used as a feature extractor for the trainable, fully connected network. [58] If a large target domain dataset is available, the full network can be trained using the pretrained weights as a starting point.

## 3.3 Evaluation metrics for imbalanced classification

When training a machine learning model, the performance metric  $P$  explained in Section 3.2.1 is used as the primary value for evaluating model performance. When choosing a performance metric, the possible real-world domain should be kept in mind. For example, classification models in medicine should utilize different metrics that classification models used in industrial computer vision applications.

This chapter illustrates the theory behind popular performance metrics, starting from primary confusion matrix metrics and moving on to more sophisticated secondary metrics derived from these. Finally, the useful connection between precision and Bayesian prob-

ability is discussed.

### 3.3.1 Confusion matrix metrics

When evaluating classification performance using categorical variables, the output of our model is either equal ( $y = \hat{y}$ ) or unequal ( $y \neq \hat{y}$ ) with the true value. Several metrics based on this comparison are often derived from the *0-1-loss*, which is defined by [57]

$$L_{0-1}(y, \hat{y}) = \begin{cases} 0, & \text{if } y = \hat{y} \\ 1, & \text{if } y \neq \hat{y}. \end{cases} \quad (3.13)$$

The *0-1-loss* is fairly simple and provides a good optimization target for classification problems, but it turns out to be hard to optimize and use in practice [59]. Probably the most common metric in evaluating machine learning models derived from the 0-1-loss is the *accuracy score*, defined by the sum of correct predictions  $y = \hat{y}$  over the total amount of samples

$$\text{accuracy (acc)} = \frac{\text{number of correct predictions}}{\text{total samples}} = 1 - \frac{1}{N} \sum_{n=1}^N L_{0-1}(y_n, \hat{y}_n) \quad (3.14)$$

which can also be used to define *classification error* by

$$\text{error (err)} = 1 - \text{acc} = \frac{1}{N} \sum_{n=1}^N L_{0-1}(y_n, \hat{y}_n) \quad (3.15)$$

where in both  $N$  is the total amount of samples and  $L_{0-1}(y_n, \hat{y}_n)$  is the *0-1 loss*.

When evaluating models using hierarchical taxonomic data, Årje et al. [4] recommend using *context-sensitive error* metric described by Verma et al. [60], and modifying it to produce a *level-aware context-sensitive error* (LCSE) metric. LCSE is defined by

$$\text{LCSE} = \frac{1}{N} \sum_{n=1}^N \frac{1}{H_n} L(y_n, \hat{y}_n) \quad (3.16)$$

where  $L(y_n, \hat{y}_n)$  is now the height of the deepest common ancestor if  $y \neq \hat{y}$ , and 0 otherwise. Because the highest available hierarchy level can vary,  $y$  refers to the most accurate level of hierarchy and  $H_n$  is the total hierarchical level count for the observation in question. LCSE takes class hierarchies better to account, especially if the training data contains samples with varying hierarchies [4].

Accuracy metrics derived from the 0-1 loss are not always the best possible measures for determining model performance. In many cases, the consequences of a missed positive are more costly than a false positive prediction in a binary classification case. A popular example is the case of cancer detection in a patient. If a sick patient's cancer diagnosis

goes undetected, it can have far more severe consequences than a healthy patient being diagnosed sick. This calls for classifications to be arranged in a better way than just comparing exactly correct classifications  $y = \hat{y}$ .

### Confusion matrix

A well-known method for binary classification with two labels  $\mathcal{Y} = \{0, 1\}$ , with origins in information retrieval is the use of a *contingency table* for different classes of predictions. In information retrieval the results are divided to four parts: the *retrieved* and *not retrieved*, meaning the classes of the predictions being either 1 or 0, and the *relevant* and *not relevant*, meaning the true class of the prediction being either 1 or 0. [61] This table can be extended to a more general case by interpreting retrieval as the predicted label, and relevance as the true label of a sample. The combinations of correct and incorrect predictions form a  $2 \times 2$  matrix presented in table 3.2 and referred to as a *confusion matrix*. [15, 17]

**Table 3.2.** Confusion matrix for binary classification

		True	
		$y = 1$	$y = 0$
Predicted	$\hat{y} = 1$	True Positives (TP)	False Positives (FP)
	$\hat{y} = 0$	False Negatives (FN)	True Negatives (TN)

The confusion matrix can be extended for multi-class classification. With  $n$  labels the result is a  $n \times n$  matrix (Table 3.3) with all possible combinations of true and predicted labels for each class. Multi-class classification can be reduced to binary classification separately for each class, by forming the confusion matrix in a "one-vs-rest" fashion seen in table 3.4. The true labels are counted for  $y = l_i$  and  $y \neq l_i$ , where  $l_i$  is the label under inspection. The one-vs-rest method is useful for calculating evaluation metrics for multi-class models.

**Table 3.3.** Multi-class confusion matrix. Nominal classes represented by letters  $A, B, C$ . Diagonal true positives are bolded. Non-diagonal cells contain the two possible error interpretations for each class.

		True		
		$y = A$	$y = B$	$y = C$
Predicted	$\hat{y} = A$	<b>TP<sub>A</sub></b>	FP <sub>A</sub> (FN <sub>B</sub> )	FP <sub>A</sub> (FN <sub>C</sub> )
	$\hat{y} = B$	FP <sub>B</sub> (FN <sub>A</sub> )	<b>TP<sub>B</sub></b>	FP <sub>B</sub> (FN <sub>C</sub> )
	$\hat{y} = C$	FP <sub>C</sub> (FN <sub>A</sub> )	FP <sub>C</sub> (FN <sub>B</sub> )	<b>TP<sub>C</sub></b>

The multi-class confusion matrix seen in table 3.3 contains rich information on different kinds of classification errors. The matrix diagonal contains the true positive (TP) values for each class, which are also the true negatives (TN) for the other classes. The non-diagonal values contain the classification error counts - each value can be interpreted

either as false positive (FP) or false negative (FN) values of the class on the corresponding row and column.

**Table 3.4.** Multi-class confusion matrix as binary one-vs-rest matrix with class  $A$  as the class under inspection

		True	
		$y = A$	$y = \{B \text{ or } C\}$
Predicted	$\hat{y} = A$	True Positives (TP)	False Positives (FP)
	$\hat{y} = \{B \text{ or } C\}$	False Negatives (FN)	True Negatives (TN)

The confusion matrix is a powerful way of describing classification results. Several metrics can be derived from the confusion matrix values, that better describe the true predictive value of the classifier [15, 18]. For example, take a highly imbalanced test dataset with a total of 100 samples: 99 samples of class  $y = 0$  and 1 sample of class  $y = 1$ . If we build a classifier that for any given  $x$  predicts  $h(x) = 0$ , we are able to correctly classify all of the 99 samples of class 0, giving us an accuracy of 0.99. Although the accuracy is high, the model is still poorly formed since it never predicts the positive class 1. If we had a doctor tasked with predicting cancer in patients saying all patients she encounters are healthy, she would be right in 99% of the cases but would still be a horrible doctor. We can say that the *true positive rate* (TPR) of the model is 0.00, meaning the correctly predicted positive labels over the total amount of positive labels. TPR can also be called *recall* or *sensitivity* [18]. Similar to TPR, we can calculate the *true negative rate* (TNR, inverse recall or specificity) that is the opposite of TPR. TNR calculates the correct negative predictions over all truly negative samples, giving an accuracy of a negative prediction being true [16].

The above example raised a problem with a classifier not being able to recall the true positive samples from the test set. Another common metric is *precision*, or *positive prediction value* (PPV) which calculates the true amount of positive predictions over all samples predicted positive [18]. For example, take 90 negative samples and 10 positive ones. If our model chose 10 positive predictions and 90 negative ones by random and somehow accomplish choosing 5 of the positive predictions right, we would have an accuracy of 0.9, with a model working somewhat unreliably. Thankfully the precision score of 0.5 will indicate that something is not right in the classifier. The formulas for calculating the accuracy, recall, inverse recall, and precision from the confusion matrix values are presented in table 3.5.

### Imbalanced classification

The use of these standard metrics has also been criticized due to them being biased for positive values [18]. If we take the cancer-detecting doctor example above with 99 negative and a single positive sample and switch the classes, we would get a recall score of 0.99 (since 99% of the positive predictions are correct) and accuracy similar to this. In the case of recall, it is necessary to know the difference between whether to use recall

**Table 3.5.** Common classification evaluation metrics [16, 18]

Metric	Formula	Description
Accuracy (ACC)	$\frac{TP+TN}{TP+FP+TN+FN}$	Correct predictions over total number of samples
True positive rate (TPR), Recall, Sensitivity	$\frac{TP}{TP+FN}$	Probability of detection
True negative rate (TNR), Inverse recall, Specificity	$\frac{TN}{TN+FP}$	Probability of correct rejection
Positive prediction value (PPV), Precision	$\frac{TP}{TP+FP}$	Probability of positive prediction to be correct
Negative prediction value (NPV)	$\frac{TN}{TN+FN}$	Probability of negative prediction to be correct

or inverse recall. In other words, depending on the classification task, it should be well known which metric, false alarms (FP) or misses (FN), is more important.

With imbalanced classification, it is common to have the minority class set as the positive class. With significant class imbalance, the negative class metrics (TNR, NPV) often get high values close to 1. For example, with 100 positive samples and 99 900 negative samples, the model most likely ends up learning to predict the negative class, resulting in low recall and precision scores. Due to the imbalance, it is unlikely that the model predicts large amounts of false positives, resulting in a TNR close to 1. Similarly, due to the small number of positive values, the maximum amount of false negatives is at most 100 with our example case, resulting in an NPV score close to 1.

This characteristic of imbalanced classification leads to a situation where the positive minority class performance describes the model's overall performance better. Well-known metrics that combine positive performance metrics for calculating single score metrics for model performance are, for example, the F-measure and G-mean. Other metrics, such as the Matthews Correlation Coefficient (MCC), attempt to take also the negative performance metrics to account to provide a more balanced score [18]. However, when the imbalance between positive and negative classes is significant, as it often is in a multi-class setting, the negative performance metrics get values close to 1. This makes using the positive metrics, recall (TPR) and precision (PPV), simpler and more useful for most use cases.

The metrics that can be straight derived from the confusion matrix (TPR, TNR, PPV, NPV) are referred to as *primary metrics* in this thesis. Other metrics that are further derived from these values by combining them in some way are referred to as *secondary metrics*.

### 3.3.2 Averaging multi-class classifiers

Micro- and macro-averaging refers to two different methods of calculating a single metric for a multi-class classifier [19]. Micro-averaging calculates metrics by using cumulative

sums of confusion matrix values over all classes. Macro-averaging calculates a metric for each class separately, with the one-vs-rest method, and averages this score over the number of classes.

Micro-averaged precision is calculated with

$$micro = \frac{\sum_{c=1}^C TP_c}{\sum_{c=1}^C (TP_c + FP_c)} \quad (3.17)$$

where TP for example is the number of true positives for class  $c$ . Similarly, macro-averaged precision is calculated with

$$macro = \frac{\sum_{c=1}^C \frac{TP_c}{TP_c + FP_c}}{C} \quad (3.18)$$

where  $C$  is the number of classes similarly as above. [19]

Micro-averaging is generally not useful in a multi-class application where each sample has only one label. When calculating micro-averages for scores like accuracy, precision and recall, due to the cumulative sum over classes, FP and FN will result in the same number: the number of errors. This leads to accuracy, precision, and recall being the same when micro-averaged over all classes. Macro-averaging weighs the class metrics evenly, giving equal importance to all classes regardless of size, and resulting in a fairly good overall performance metric. However, micro-averaging over cross-validation folds has been argued to be a good practice, resulting in a less biased average for a cross-validated metric [62]. Averaging over cross-validation folds in a multi-class setting is hard to find from the literature, and methods for this are presented later in chapter 5.

Where macro-averaging refers to taking the arithmetic mean of a performance metric over the classes, slightly different single-score metrics can be produced by calculating alternatively the *geometric mean* or *harmonic mean* instead. Because the most common performance metrics have values between 0 and 1, the values can be interpreted as normalized values. It has been argued that when averaging normalized values, arithmetic mean can produce incorrect conclusions, and geometric mean should be used instead [63].

### 3.3.3 F-measure, G-mean

The *F-measure* or F-score is a secondary performance metric derived from the confusion matrix values. The general version of the metric, the  $F_\beta$ -score, is calculated from recall and precision with the equation

$$F_\beta = \frac{(1 + \beta^2) \cdot Recall \cdot Precision}{(\beta^2 \cdot Recall) + Precision} \quad (3.19)$$

where  $\beta$  can be interpreted as the importance of precision compared to recall. A common  $\beta$  value is 1, producing the F1-score, which the terms F-measure or F-score usually refer to. [15] In this case the equation for F1-score is

$$F_1 = \frac{2 \cdot \text{Recall} \cdot \text{Precision}}{\text{Recall} + \text{Precision}} = \frac{2 \cdot TP}{2 \cdot TP + FN + FP} \quad (3.20)$$

which corresponds to the harmonic mean of precision and recall. F1-score is considered a fairly good metric for evaluating imbalanced data by combining precision and recall to evaluate the effectiveness of positive class detection [15, 18]. A problem identified with F1-score is that it is biased for the positive class by ignoring TN values. [18] Often, an F-measure with a custom  $\beta$  should be considered since the F1-score gives equal importance to both precision and recall, which in many situations might not be the desired case [64]. Using F1-score as a target metric results in a good performance with the positive class but ignores the performance of the negative class, which usually is a combination of "the other" classes in a multi-class problem. This is not a problem if the choice of the positive class has been made well, and positive class performance is desired. With imbalanced data, usually present in a "one versus the others" setting, the minority class should be chosen as the positive class to prevent problems described in Section 3.3.1.

A metric that attempts to maximize the accuracy of both positive and negative class is the *G-mean* defined by the equation

$$G\text{-mean} = \sqrt{\text{Recall} \cdot \text{Specificity}} \quad (3.21)$$

which takes also the negative class' recall (specificity) to account [65].

The G-mean essentially balances TPR and TNR, becoming useful if the specificity values vary a lot. If the class imbalance is high, it is generally better to use just TPR as a metric since with TNR values close to one, the G-mean acts just as a squared TPR value.

G-mean can be easily confused with *G-measure*, since the terms are often used interchangeably in the literature. In this thesis the term *G-measure* refers to the geometric mean of recall and precision

$$G\text{-measure} = \sqrt{\text{Recall} \cdot \text{Precision}} \quad (3.22)$$

which is similar to the F-measure, using geometric mean instead of harmonic mean [18]. The G-measure performs very similarly to the F1-score since both calculate the mean of precision and recall, but with different Pythagorean means. Since precision and recall have values between  $[0, 1]$ , the difference between F1-score and G-measure is even at its largest very small. For this reason, G-measure is not discussed further, and focus is given on the F1 score instead.

As with other metrics, it is possible produce a single-score metric for a multi-class classifier by averaging the F1-score over classes. This is done by applying equation 3.20 to

either equation 3.17 for micro-averaging or equation 3.18 for macro-averaging. Again, using geometric mean for averaging can be more justified than arithmetic mean, producing more reliable conclusions.

### 3.3.4 Bayesian probability

Bayesian probability is highlighted as a performance metric that is useful and intuitive to understand, especially when evaluating classifiers used in automated biological monitoring. Probabilities can generally be divided into two classes: *frequentist probability* and *Bayesian probability*, where Bayesian probability discusses the *degree of belief* of a classification instead of the probability of event occurrence, associated with frequentist probability [50, p. 53]. This is often the way humans intuitively interpret the outputs of automatic classification models, making it suitable as a performance metric that is evaluated by non-machine-learning professionals.

The Bayes rule is defined with the equation [56, p. 29]

$$p(A|B) = \frac{p(B|A)p(A)}{p(B)} \quad (3.23)$$

where  $p(A|B)$  is the conditional probability of A being true given that B is true, or the *posterior probability*. When discussing a classification problem, equation 3.23 can be interpreted as

$$p(y = c|\hat{y} = c) = \frac{p(\hat{y} = c|y = c)p(y = c)}{p(\hat{y} = c)} \quad (3.24)$$

where the probability  $p(y = c|\hat{y} = c)$  is the conditional probability of a label with the value  $c$  being true, given a classifier output suggesting this class. This illustrates the Bayesian probability being interpreted as a degree of belief, or certainty of a proposition being true. It is possible to see from the definition of Bayesian probability that it corresponds to the precision (PPV) value calculated from the confusion matrix. Although intuitive, notion was hard to find from the literature. This connection will be thus proved here.

For the other terms in the Bayes theorem,  $p(\hat{y} = c|y = c)$  is the *likelihood* of a detection: the probability that a label is classified correctly (detected), which is the same as TPR. The second term in the numerator,  $p(y = c)$ , corresponds to the *prior probability* of a class, that can be calculated from an assumed probability distribution or separately for each class with

$$p(y = c) = \frac{Pos}{Pos + Neg} = \frac{TP + FN}{TP + FN + FP + TN} \quad (3.25)$$

where  $Pos$  and  $Neg$  are the amounts of positive and negative samples.

Finally, using the law of total probability we get

$$p(\hat{y} = c) = \sum_i^C p(\hat{y} = c|y = c_i)p(y = c_i) \quad (3.26)$$

which is the probability of a classification being made. Using table 3.4 as a reference, it can be seen that the sum in equation 3.26 consists of values that are either  $y = c$  when  $c = c_i$ , or  $y \neq c$  when  $c \neq c_i$ . The associated probabilities are then  $p(y = c)$  and  $1 - p(y = c)$  and the equation can be simplified as

$$p(\hat{y} = c) = p(\hat{y} = c|y = c)p(y = c) + p(\hat{y} = c|y \neq c)(1 - p(y = c)) \quad (3.27)$$

where  $p(\hat{y} = c|y \neq c)$  corresponds to the *false positive rate* or  $1 - \text{TNR}$ .

Now the equation 3.24 can be shown in terms of the confusion matrix metrics as

$$\begin{aligned} p(y = c|\hat{y} = c) &= \frac{TPR \cdot p(y = c)}{TPR \cdot p(y = c) + (1 - TNR) \cdot (1 - p(y = c))} \\ &= \frac{\frac{TP}{Pos} \cdot \frac{Pos}{Pos+Neg}}{\frac{TP}{Pos} \cdot \frac{Pos}{Pos+Neg} + FPR \cdot \frac{Neg}{Pos+Neg}} \\ &= \frac{TP}{TP + FPR \cdot Neg} \\ &= \frac{TP}{TP + FP} = PPV \end{aligned} \quad (3.28)$$

We see that the Bayesian probability is the same as precision, which makes intuitively sense since precision is the probability of a positive prediction to be correct. This makes precision an important metric when evaluating classification models, especially ones that need a metric for the confidence of a classification to be correct.

### 3.4 Loss functions for imbalanced data

One of the most significant differences between traditional and machine learning algorithm optimization is that while traditional optimization seeks to minimize or maximize an loss (objective) function  $J(\theta)$ , machine learning algorithms seek to minimize (maximize) a performance metric  $P$  that is not directly available during the optimization [50]. This characteristic of statistical learning poses problems that can be amplified with an unbalanced dataset.

Unbalanced data can lead to problems where the data-generating and empirical distribu-

tions have large differences. If the original distribution  $p_{data}$  is highly unbalanced and the sample size is not sufficient, two problems can arise: the empirical distribution might not necessarily match the data-generating distribution, or the number of samples representing smaller classes might not be large enough for the model to learn class-distinguishing features correctly.

This section introduces probably the most used loss-function for classification problems, the *cross-entropy loss*, as well as two other loss-functions designed especially for imbalanced classification problems: the *focal loss* and *class-balanced loss*.

### 3.4.1 Cross-entropy loss

Information theory defines the *entropy* of a random variable  $X$  to be

$$H(X) = H(p) = - \sum_{i=1}^C p(x_i) \log p(x_i) \quad (3.29)$$

where  $p(x_i)$  refers to the probability of the  $i$ :th possible outcome of  $X$ , analogous to the probability distribution of  $X$  [66]. If base-2 logarithm is used, the units are called *bits* and with natural logarithm *nats*. Similar definition can be used to calculate the entropy between two random variables and their distributions  $p$  and  $q$  with

$$H(p, q) = - \sum_{i=1}^C p(x_i) \log q(x_i) \quad (3.30)$$

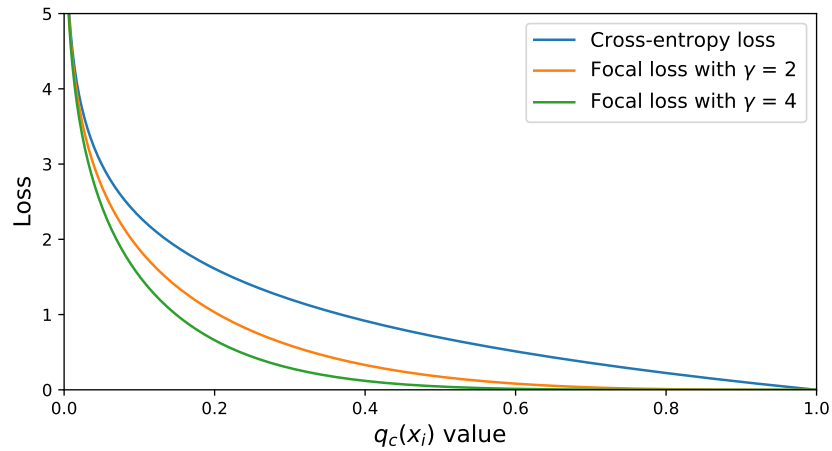
which is called the *cross-entropy* between distributions. [56] The cross-entropy is a commonly used loss function for calculating the performance of a classification in a differentiable manner. If an classifier outputs a probability distribution  $f(\mathbf{x}) = q(y|\mathbf{x})$  as explained in Section 3.2.3, for the input  $\mathbf{x}$  and the true mapping for the input is the distribution  $h(\mathbf{x}) = p(y|\mathbf{x})$ , the cross-entropy loss is can be derived from equation 3.30 to be

$$CE(\mathbf{x}) = - \sum_{i=1}^C p(y_i|\mathbf{x}) \log q(y_i|\mathbf{x}) \quad (3.31)$$

where  $q(y_i|\mathbf{x})$  is the probability distribution value corresponding to the  $i$ :th class. If the true class for input  $\mathbf{x}$  is  $c$ , the true distribution  $p(y|\mathbf{x})$  is an *one-hot* -vector with values

$$p(y_i|\mathbf{x}) = \begin{cases} 1, & \text{if } y_i = c \\ 0, & \text{if } y_i \neq c \end{cases}$$

that naturally sum up to 1. The cross-entropy loss in its general form can be referred to as *categorical cross-entropy*, and as *binary cross-entropy* in the special case  $C = 2$ . With



**Figure 3.3.** Cross-entropy and focal loss as a function of classifier output probability score. Focal loss outputs lower loss scores for well-classified samples, giving more attention to harder classes. Modified from [32]

one-hot -vectors, the cross-entropy loss is possible to write in a simplified form

$$CE(\mathbf{x}) = -\log q_c \quad (3.32)$$

where we define  $q_c$  as the probability  $q_c = q(y = c_{true} | \mathbf{x})$  where  $c_{true}$  is the true class.

Often in cost-sensitive learning the cross-entropy loss is weighted with a additional term  $\alpha$  to produce a loss-function more suitable for unbalanced datasets [14, 32]. The term can be for example the inverse frequency of the classes or the normalized frequency subtracted from one.

The cross-entropy loss is probably the most common loss function for classification models outputting probability density functions over classes, due to its differentiability and high cost for incorrect classifications. Often the cost function to be minimized is calculated by averaging the cross-entropy among a batch of samples.

### 3.4.2 Focal loss

Cross-entropy loss is generally a good loss-function due to the logarithmic function penalizing classifiers outputting low probability scores for true labels. Figure 3.3 shows the value of the cross-entropy loss function with different probability scores. Cross-entropy loss maintains a fairly high loss value up until the probability reaches 1, resulting in large optimization steps also for well-classified ( $q \geq 0.6$ ) samples.

The classic cross-entropy loss can overwhelm the classification of harder classes, espe-

cially when the data is unbalanced, and the well-classified samples are more numerous than the harder samples. Lin et al. have proposed using a *focal loss* function to address this problem and give more focus on the harder examples. Focal loss adds a factor  $(1 - q)^\gamma$  to the cross-entropy loss, resulting in a lower loss for well-classified samples. For example, a sample classified with a probability score of  $q_c = 0.9$  has a focal loss score 100 times lower than cross-entropy. [32] The difference between focal loss and cross-entropy loss can be seen in figure 3.3. It can be seen that focal loss keeps penalizing classifications with poor confidence, but for well-classified samples with a high probability score, the loss decreases faster than cross-entropy. This gives more focus to the rare samples typically having poor classification confidence due to insufficient training data, as the losses well-classified samples do not dominate the total loss and the gradient during optimization.

The focal loss is defined with the equation

$$FL(\mathbf{x}) = - \sum_{i=1}^C (1 - q(y_i|\mathbf{x}))^\gamma (p(y_i|\mathbf{x}) \log q(y_i|\mathbf{x})) \quad (3.33)$$

which can be simplified similarly to equation 3.32

$$FL(q_c) = -(1 - q_c)^\gamma \log q_c \quad (3.34)$$

when the true distribution  $p(y|\mathbf{x})$  has the value 1 for the true class and 0 for others. [32]

The focusing parameter  $\gamma$  affects how much the well-classified samples are down-weighted. With value  $\gamma = 0$  focal loss is equivalent to cross-entropy, and with higher values the well-classified samples get lower loss values. Lin et al. [32] describe that the value  $\gamma = 2$  works well in practice, but for object detection tasks values  $\gamma = [0.5, 5]$  work relatively well. In practice Lin et. al. recommend adding a balancing parameter  $\alpha$  to the equation

$$FL(q_c) = -\alpha_c (1 - q_c)^\gamma \log q_c \quad (3.35)$$

where  $\alpha_c$  corresponds a balancing factor for class  $c$ . The factor  $\alpha_c$  can be for example the inverse frequency of the class, which results the focal loss giving more attention to rare classes. [32]

### 3.4.3 Class-balanced loss

Class-balanced loss (CB loss) is a loss function presented by Cui et al., designed for long-tailed datasets. Compared to focal loss, which reduces the weight of well-classified samples, class-balanced loss reduces the weight of classes with a high *number* of samples. The weighing is done by a factor of  $(1 - \beta^n)/(1 - \beta)$  where  $n$  is the number of available samples, and  $\beta$  is a hyperparameter. Cui et al. argue that when the number of samples in a class increases, the benefit for training diminishes because it is more likely

that the new sample is very similar to an existing sample in the dataset.

Cui et al. propose calculating an *effective number*  $E_{n_c}$  for each class  $c$

$$E_{n_c} = (1 - \beta^n)/(1 - \beta) \quad (3.36)$$

where  $\beta$  is calculated from  $\beta = (N - 1)/N$ ,  $N$  being the size of the dataset containing *all possible data* of the class. This number is hard to find empirically for each class, so the authors suggest using  $\beta$  as a hyperparameter, same for all classes in the dataset. The final loss consists of a loss function  $L(f(\mathbf{x}), y)$  that is weighed inversely proportional to the effective number producing the class-balanced loss

$$CB(f(\mathbf{x}), y) = \frac{1}{E_{n_c}} L(f(\mathbf{x}), y) = \frac{1 - \beta}{1 - \beta^{n_c}} L(f(\mathbf{x}), y) \quad (3.37)$$

Class-balanced loss can be used with, for example, the cross-entropy loss (CE) or focal loss by substituting the loss function  $L$  in equation 3.37 accordingly.

## 3.5 Sampling methods for imbalanced data

In addition to designing suitable loss functions, problems with imbalanced datasets can be addressed by modifying the training dataset itself. Several methods for this exist, but generally different *resampling methods* and *data augmentation* have proved to be useful [24, 25, 26, 28]. Sampling methods focus on artificially altering the training dataset  $S$  and its empirical distribution  $\hat{p}_{data}$  to acquire a more general model. Data augmentation alters the data itself to produce more samples that could theoretically be sampled from the underlying distribution  $p_{data}$  thus training the model with additional points of reference. The notation  $S'$  is used here to refer to the altered dataset and can be produced either by sampling, augmentation, or both.

When altering a dataset, it is crucial to ensure that the distribution of the new dataset  $S'$  follows the data-generating distribution  $p_{data}$ . If the distributions differ, the empirical distribution might represent some other data-generating distribution, resulting in problems discussed in 3.2.1.

### 3.5.1 Resampling methods

Sampling methods can generally be divided into two classes: undersampling and oversampling. When undersampling a dataset, some samples in the majority class are discarded to produce a more even distribution. Oversampling does the opposite by increasing the amount of the minority class samples by replicating them.

Several sophisticated methods exist for both under- and oversampling, one of the most popular ones being SMOTE [29], which oversamples the minority classes by adding new synthetic data between a sample and one of its nearest neighbors. In addition to re-

sampling, SMOTE performs augmentation to the images by essentially creating a new sample. New samples using this method are added to the new dataset until a sufficient balance is achieved.

A study by Batista, Prati and Monard [27] suggests that just using simple random over- and undersampling can produce excellent results when compared to more sophisticated methods. Here random over- and undersampling refers to randomly choosing samples in the original dataset  $S$  to be either replicated or discarded, ultimately producing a new dataset  $S'$  with better balance, without augmenting the original samples. This ensures that the image representations are from the original data-generating distribution, but the dataset distribution between classes can represent a more balanced sample.

### 3.5.2 Data augmentation

Data augmentation has proved to be a good method for addressing problems with insufficient data, especially when used with neural networks [24, 67]. Data augmentation can fix problems present both with too small datasets as well as with imbalanced datasets and is generally considered as a best practice used in several benchmark-breaking classifiers [68]. Augmentation can result in more robust and general classifiers by presenting the model new samples augmented with realistic modifications [67].

Several augmentation methods exist, such as SMOTE discussed earlier. The augmentation performed by SMOTE differs from most commonly used in image classification since SMOTE performs augmentation on the feature-level, producing a point between two existing image points in a very high-dimensional space [29]. This method is acknowledged to be a valid method of augmentation [25, 68], but since the augmentation is performed in feature-space, the modifications differ from what are usually considered as "augmentations" by humans, referred to as a *data-warping* approach [68].

Different data-warping augmentations can produce a variety of new samples for the training set. Arguably the most popular methods of augmentation for image data are performing different geometric distortions and histogram modifications. Examples of geometric modifications are shearing, rotating, and random cropping of the images, when the histogram and color modifications can, for example, change the contrast of an image, perform sharpening or blurring, or imitate different lighting conditions by slightly changing the white-balance of an image. [26]

## 4 METHODS

This chapter along with chapters 5 and 6 presents the main findings of this thesis: the comparison between different evaluation metrics, and experiments on different loss-functions and resampling methods. The experiments were done by building classification models trained with benthic macroinvertebrate data collected by the Finnish Environment Institute (SYKE). This chapter focuses on the background methods for the experiments, and the main chapters for both performance metrics (Chapter 5) and improvement methods (Chapter 6) discuss the main findings and results in depth.

This chapter discusses the data used and the methods for building the reference model. In Section 4.1, the benthic macroinvertebrate dataset is described, with the challenges associated with hierarchical and unbalanced data. Section 4.2 discusses the training and implementation details of the reference model used for both metric evaluation, as well as for comparing classification improvements.

### 4.1 Data

The data used for the experiments was compiled by the Finnish Environment Institute, and is publicly available at <https://etsin.fairdata.fi/> under the name "FIN-Benthic 2". The full dataset consists of 126 lotic macroinvertebrate taxa of over 2.6 million images; however, a subset of 460004 images from 39 taxa classes and 9631 individual specimen was used in [4]. Images were collected using a system described by Raitoharju et al. [8], consisting of two cameras which automatically take images of the macroinvertebrate while it sinks in an alcohol-filled test cuvette. Total images per sample depends on the size and weight of the macroinvertebrate due to heavier samples sinking faster. In addition to the dataset being a subset of the full data, the the maximum number of images per specimen was limited to 50 [4].

The data consists of 39 taxonomic groups classified on different levels of taxonomic resolution. In total the data has 7 orders, 23 families, 30 genera and 26 species level classifications. The 39 taxa classes were classified on the deepest available taxonomic rank. This means that depending on the taxa, the most accurate classification is not necessary on species level but can be genus or family instead. The hierarchical differences of the classes can be seen in figure 4.1. [4] The 39 taxa groups, with their respective order, family, genus and species classifications are shown in appendix A.1.

The 39 taxa classes have been labeled numerically in alphabetical order for convenience.

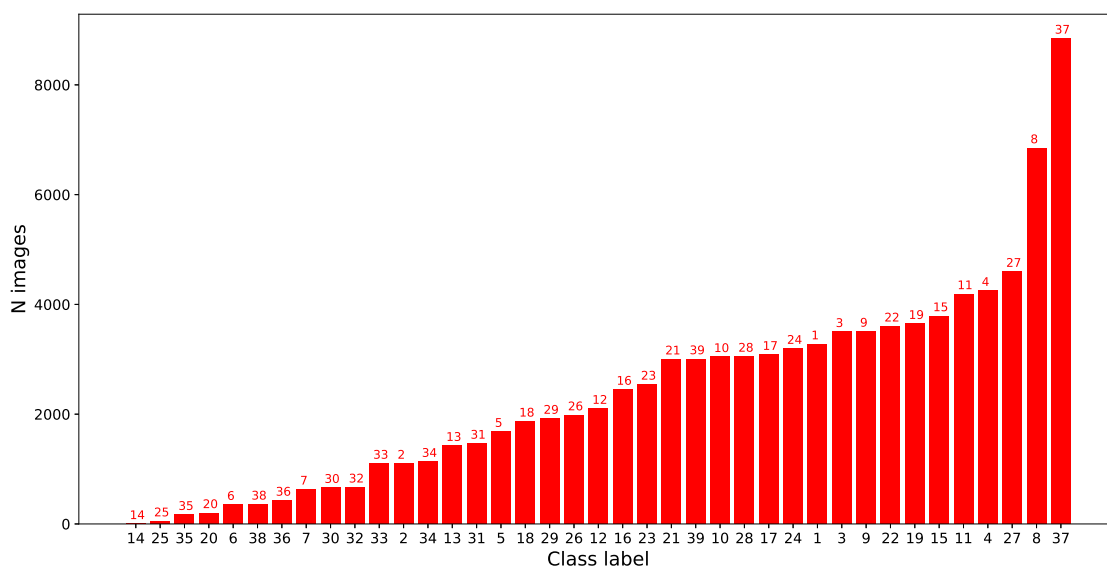


**Figure 4.1.** Taxonomic resolution of the image data. The area of each slice corresponds to the size of the taxonomic group [4]

**Table 4.1.** Taxa labels

Label	Taxa	Label	Taxa	Label	Taxa
1	Agapetus sp.	14	Hydropsyche saxonica	27	Neureclipsis bimaculata
2	Ameletus inopinatus	15	Hydropsyche sitalai	28	Oulimnius tuberculatus
3	Amphinemura borealis	16	Isoperla sp.	29	Oxyethira sp.
4	Baetis rhodani	17	Kageronia fuscogrisea	30	Plectrocnemia sp.
5	Baetis vernus group	18	Lepidostoma hirtum	31	Polycentropus flavomaculatus
6	Capnopsis schilleri	19	Leptophlebia sp.	32	Polycentropus irroratus
7	Diura sp.	20	Leuctra nigra	33	Protonemura sp.
8	Elmis aenea	21	Leuctra sp.	35	Sialis sp.
9	Ephemerella aurivillii	22	Limnius volckmari	36	Rhyacophila nubila
10	Ephemerella mucronata	23	Micrasema gelidum	36	Silo pallipes
11	Heptagenia sulphurea	24	Micrasema setiferum	37	Simuliidae
12	Hydraena sp.	25	Nemoura cinerea	38	Sphaerium sp.
13	Hydropsyche pellucidula	26	Nemoura sp.	39	Taeniopteryx nebulosa

The labeling used can be seen in table 4.1. In the following figures, the taxa classes are usually referenced using their numerical labels. The class imbalance is visualized in the histogram in figure 4.2. The largest class, 37-Simuliidae, is over 90 times larger than the smallest with only total of 490 images. The large size of this class is due to it being the only taxa classified on the family level, containing a larger amount species and samples under the same label. The largest species level class is 8-Elmis aenea, which with 32398 images is 66 times larger than the smallest class. Exact sizes for all classes can be seen in table A.1.



**Figure 4.2.** Class sizes of the full dataset. The largest class (37-Simuliidae, 44240 images) is 90 times larger than the smallest (14-Hydropsyche saxonica, 490 images)

The data was further split to training (70%), test (20%) and validation (10%) sets. This split was done by random stratified sampling 10 times to create a 10-fold cross-validation dataset for training and evaluation. Splitting was done specimen-wise, so that a split contains all images of a single specimen. [4] Due to the large imbalance some of the validation splits might not necessarily contain all possible classes.

## 4.2 Reference model

To evaluate the possible improvements in the classification of rare classes, a reference classifier was trained. No special measures to account the class imbalance were taken in training the reference classifier in order to gain better insight to what methods work best for the imbalanced taxa classification problem.

A very deep convolutional neural network (CNN) model using the Inception architecture presented by Szegedy et. al. [69] was used for the reference model. Transfer learning, explained in 3.2.3, was applied by pretraining the reference model weights using the ImageNet [70] data. For the reference CNN, the classification layers in the Inception architecture were replaced with a global average pooling layer, followed by a hidden fully connected layer with 256 neurons, and ending with an softmax output layer of 39 classes. All the layers, including the convolutional layers, were made trainable due to the large availability of target domain data. The technical implementation of the neural network was done using Tensorflow [71].

The reference network was trained for 15 epochs using the Adam [72] optimization algorithm and a categorical cross-entropy loss-function. For the first 10 epochs, a learning

rate of 0.001 was used. This was further decreased according to the equation

$$LR = 0.0001 \cdot e^{0.1 \cdot (-\epsilon)} \quad (4.1)$$

where  $\epsilon$  is the epoch number. Here  $\epsilon = 1$  for the 11th epoch from the beginning of training.

These specifications were kept similar in the later models for evaluating improvements in classifying rare classes of macroinvertebrates. The architecture of the network, optimization algorithm and amount of training epochs was kept the same. Changes were made only to the loss-function and data distribution when training the later models.

The reference model has 39 output classes, corresponding to the deepest available taxonomic rank of each specimen. Since the taxa are on different taxonomic hierarchies, this produces additional challenges in classification. Classes are highly imbalanced depending on the hierarchy level, and in-class variance increases when the deepest available level is higher up in the hierarchy. In this thesis, focus is given to the classification evaluation metrics and optimization loss functions. It is desirable to partition the target space to smaller classes and to use a "flat" model with 39 outputs. This makes it possible to better distinguish the effect of loss-functions and data augmentation in the model. A single flat classifier is not necessarily the best method of classifying taxa in the most accurate manner. Årje et al. [4] describe further methods for improved classification using several classifiers on different levels of taxonomic resolution.

Unless stated otherwise, classifications are done for each image separately, using equation 3.12 for the probability vector  $p(y|\mathbf{x})$  received from the model for the given image  $\mathbf{x}$ . For some metrics a majority vote over all images of a certain specimen is used as the classification label. These are stated as the *vote* classifications in the later metrics and visualizations.

## 5 COMPARISON OF EVALUATION METRICS

It is important to use suitable evaluation metrics when judging classifier performance. In taxa recognition, the overall performance of a classifier is often presented with only accuracy, although a non-justified choice of a performance metric can produce problems discussed in Section 3.3.1. As discussed in Section 3.3.1, precision, recall, and other metrics derived from the confusion matrix can describe the performance of a classifier better than accuracy.

This chapter presents several metrics and techniques that can be used to illustrate model performance in multi-class classification, especially in an unbalanced data setting. Special attention is given to methods that are judged to be most useful for biologists and other experts in other than machine learning fields. The metrics and visualizations are chosen by ease of use and speed of evaluation in mind, so that strengths and weaknesses of the models are clear also for non-machine-learning experts using the model.

In a practical situation, an automated taxa recognition system would be used to help the work of biologists and environmental scientists, with the user having a possibility to judge whether the model outputs are valid. Especially when classifying important taxa, the way the classifier performs with these classes should be well known by the user. The user can then judge which classes can be classified reliably by the system, and which need inspection by a human expert.

The metric examples and visualizations in this section display the performance of the cross-validated reference unless stated otherwise. The sums of confusion matrix values used to calculate the micro-averaged metrics can be seen in appendix A.2. Visualizations in Chapter 6 display the performance of the model trained with the first data split; the confusion matrix values for this classifier can be seen in appendix A.3. The first Table A.2 contains the micro-averaged values over four cross-validations, resulting in values about four times larger than in Table A.3, containing only a single fold.

### 5.1 Traditional metrics for model evaluation

Perhaps the simplest way to evaluate and compare machine learning models is to produce a single score that describes the model's performance. The score can be calculated over all samples, as in accuracy, or by averaging a metric over all classes using methods described in Section 3.3.2.

**Table 5.1.** Error rate and LCSE metrics for the 4-fold cross-validated reference classifier with comparison to classifier trained with same data by Årje et al.

	Reference model		Årje et al. [4]	
	Normal	Vote	Normal	Vote
$\overline{err}$	0.074	0.044	0.114	0.131
$\overline{LCSE}$	0.045	0.023	0.052	0.070

**Table 5.2.** Micro- and macro-averaged performance of the reference classifier for each cross-validation split

Metric	Split 1		Split 2		Split 3		Split 4	
	Micro	Macro	Micro	Macro	Micro	Macro	Micro	Macro
Accuracy	0.9250	0.9250	0.9281	0.9281	0.9235	0.9235	0.9273	0.9273
Precision	0.9250	0.8515	0.9281	0.8864	0.9235	0.8453	0.9273	0.8821
Recall	0.9250	0.8208	0.9281	0.8449	0.9235	0.8109	0.9273	0.8320
F1-score	0.9250	0.8300	0.9281	0.8582	0.9235	0.8206	0.9273	0.8444

Metrics such as accuracy or classification error depend only on the total amount of samples and the amount of true predictions, and are generally useful only for evaluation between simple models. In a multi-class setting these are "class-agnostic", depending only on exact predictions not taking class imbalances to account. However, these metrics are still used fairly often: for example a study by Årje et al. [4] using the same macroinvertebrate dataset as this thesis, uses the classification error ( $err$ ) and level-aware context-sensitive error (LCSE) to evaluate model performance. For comparison, the reference network performance measured with the same metrics are presented in table 5.1. Here the mean classification error over four cross-validation splits is displayed against the ten cross-validation splits done in the study by Årje et al. The current reference model uses the deeper and more sophisticated Inception architecture than the AlexNet architecture used in Årje et al. This, as expected, results in better performance.

### 5.1.1 Cross-validation and averaging

Usually in multi-class classification, more sophisticated evaluation methods than above mentioned "class-agnostic" metrics are needed. In an unbalanced case like the classification of benthic macroinvertebrates, we want to bias the metric towards the smaller classes to take these better to account. The imbalance is large, so well performing large classes can overshadow the poor performance of several smaller classes. The overall performance of the reference classifier across all cross-validation splits can be seen in table 5.2. The table shows that micro-averaging yields the same scores for different metrics in a multi-class situation.

When calculating a single score for a classifier it is important to consider the method used to calculate the score. When cross-validation is added to the equation, unclear choices between averaging methods can result in very different single-score metrics. A

**Table 5.3.** The different multi-class cross-validation averaging combinations for F1 score. Especially cross-validated models can have ambiguous single-score metrics, most popular arguably being the macro-macro-average. The almost equal score for both micro- and macro-averages over classes are due to the cross-validation folds containing almost same amounts of true positives and negatives

Averaging over classes	Averaging over cross-validation folds	
	micro <sub>c</sub>	macro <sub>c</sub>
micro <sub>k</sub>	0.9260	0.9260
macro <sub>k</sub>	0.8447	0.8383

study by Forman and Scholz [62] argues that a common method of simply calculating the mean of a metric between splits is more biased than calculating the sum of the confusion matrix values (TP, FP, TN, FN) and using these to calculate the cross-validated score. These are analogous to macro- and micro-averaging over the cross-validation folds respectively. In a multi-class, cross-validated setting there are two possible "dimensions" of averaging: averaging over folds, and averaging over classes.

Table 5.3 illustrates that all the different methods of calculating a single metric for a cross-validated classifier yield different scores. The four single-score averages are derived from equations 3.17 and 3.18, depending on the order of "pooling" confusion matrix values either by class, produce a micro-average for each fold  $k$  ( $micro_k$ ), or by cross-validation fold, producing a micro-average for each class  $c$  ( $micro_c$ ). These can be then macro-averaged over the remaining dimension (classes or folds) to produce the *macro-micro-average*. If confusion matrix values are pooled over all classes and folds, this produces the *micro-micro-average*, and if all scores are calculated separately for each class and folds, and the arithmetic mean is calculated over both dimensions, this accounts for the *macro-macro-average*. The equations for these different averages, using the precision score as an example, are presented in equations 5.1 - 5.6.

$$micro_c = \frac{\sum_{k=1}^K TP_{c,k}}{\sum_{k=1}^K (TP_{c,k} + FP_{c,k})} \quad macro_c = \frac{1}{K} \sum_{k=1}^K \frac{TP_{c,k}}{TP_{c,k} + FP_{c,k}} \quad (5.1)$$

$$micro_k = \frac{\sum_{c=1}^C TP_{c,k}}{\sum_{c=1}^C (TP_{c,k} + FP_{c,k})} \quad macro_k = \frac{1}{C} \sum_{c=1}^C \frac{TP_{c,k}}{TP_{c,k} + FP_{c,k}} \quad (5.2)$$

$$\text{micro}_c \text{micro}_k = \frac{\sum_{k=1}^K \sum_{c=1}^C TP_{c,k}}{\sum_{k=1}^K \sum_{c=1}^C TP_{c,k} + \sum_{k=1}^K \sum_{c=1}^C FP_{c,k}} \quad (5.3)$$

$$\text{macro}_c \text{micro}_k = \frac{1}{K} \sum_{k=1}^K \text{micro}_k \quad (5.4)$$

$$\text{macro}_k \text{micro}_c = \frac{1}{C} \sum_{c=1}^C \text{micro}_c \quad (5.5)$$

$$\text{macro}_c \text{macro}_k = \frac{1}{C} \sum_{c=1}^C \text{macro}_c = \frac{1}{K} \sum_{k=1}^K \text{macro}_k \quad (5.6)$$

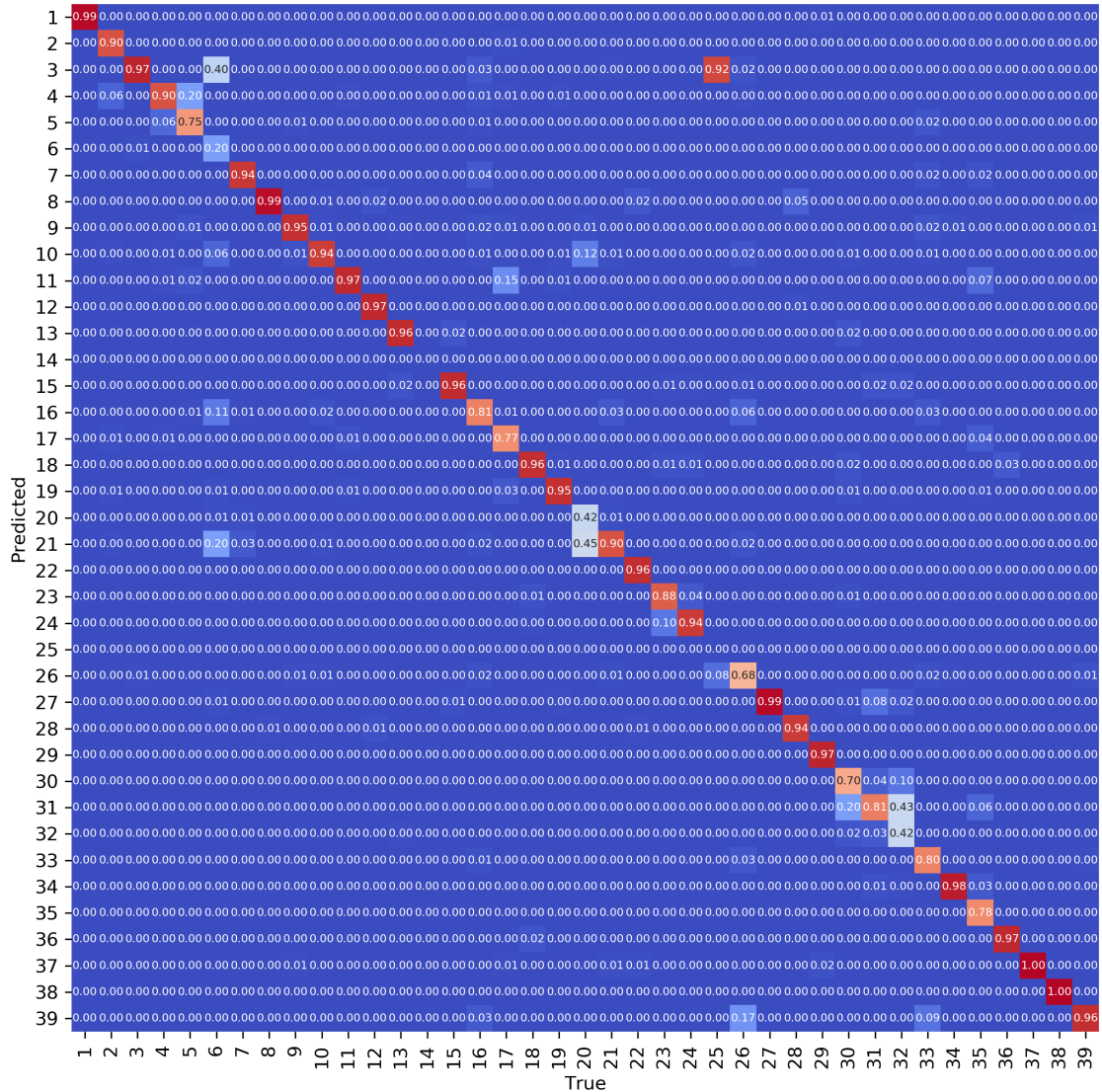
Similar calculations can be done for the F1-score, or any other confusion matrix metrics, by replacing the precision equations from equations 5.1 - 5.6 to the equation of the desired metric.

### 5.1.2 Confusion matrix and precision-recall -curve

The single-score metrics can be useful for fast evaluation of classifiers, but a more detailed inspection can be done by examining the confusion matrix, seen in figure 5.1. This confusion matrix displays the scores for the model trained and tested with the first cross-validation split. For each class, true positives can be counted from the diagonal, false positives from rows and false negatives from columns. Normalized confusion matrix columns sum to 1 and represent the distribution of the ground truth positive values across the false negative values of all classes. Normalization is done over columns, because columns represent the ground truth values, and can be seen as a probability distribution over all possible rows.

When classifying benthic macroinvertebrates, we are interested in two things: how well rare classes are detected (recall) and how reliable a positive detection is (precision). If the confusion matrix column values are spread across several classes, like in figure 5.1 the classes *6-Capnopsis schilleri* and *20-Leuctra nigra* are, the *precision* of the class is poor. Especially for class 6 the classifier reliability is poor since the values are distributed over several classes. As for class 20, the values are often misclassified to 21, making the situation is a bit better for the biologist, since one can be fairly sure the true class of the sample is one of these two.

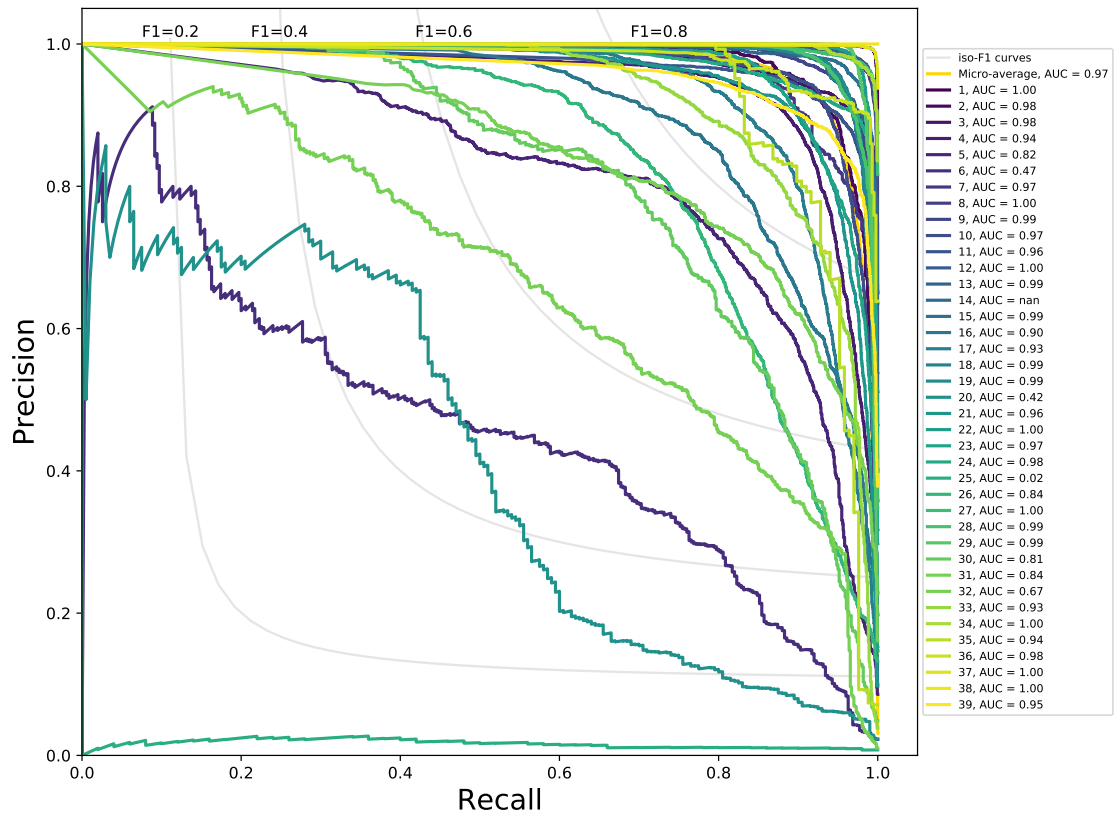
Along with the confusion matrix, a very popular method of displaying the performance of a classifier trained with imbalanced data is the precision-recall (PR) curve. The precision-recall curve calculates the precision and recall values for different classification thresholds ranging from 0 to 1 [73]. Previously the choice of each sample was done by using the most likely class, but precision-recall curve takes to account all possible thresholds a decision can be made. Usually as the threshold is increased so that more samples are detected, it affects the precision negatively. An optimal classifier will have a high precision also for higher recall values. In a multi-class setting, the different thresholds are calculated for each class using a one-vs-rest method. Each class is separately treated as the positive



**Figure 5.1.** Normalized confusion matrix of the reference model on the test set 1. Column sums are 1.00 and correspond to the class positive amounts seen in appendix A.3 . When the values are concentrated on the diagonal, model performs well. In this case, for example, the class 20-*Leuctra nigra* is often classified as class 21, which can be seen from the non-diagonal values.

class, while other classes combined produce the negatives. For each threshold value, the precision and recall values are calculated by setting values above the threshold as predicted positives and under as predicted negatives. In a practical setting, because of the nature of one-vs-rest classification, choosing a certain threshold is not straightforward, and thus the precision-recall curve should be used for general performance evaluation and not for choosing a certain threshold for inference.

Figure 5.2 illustrates the multi-class precision-recall plot for the reference classifier trained with the first dataset. Classes with poor performance can be easily detected, but the plot itself becomes fairly hard to read with larger amount of classes. In general, the precision-recall curve is one of the most informing methods of describing model performance and is widely used.

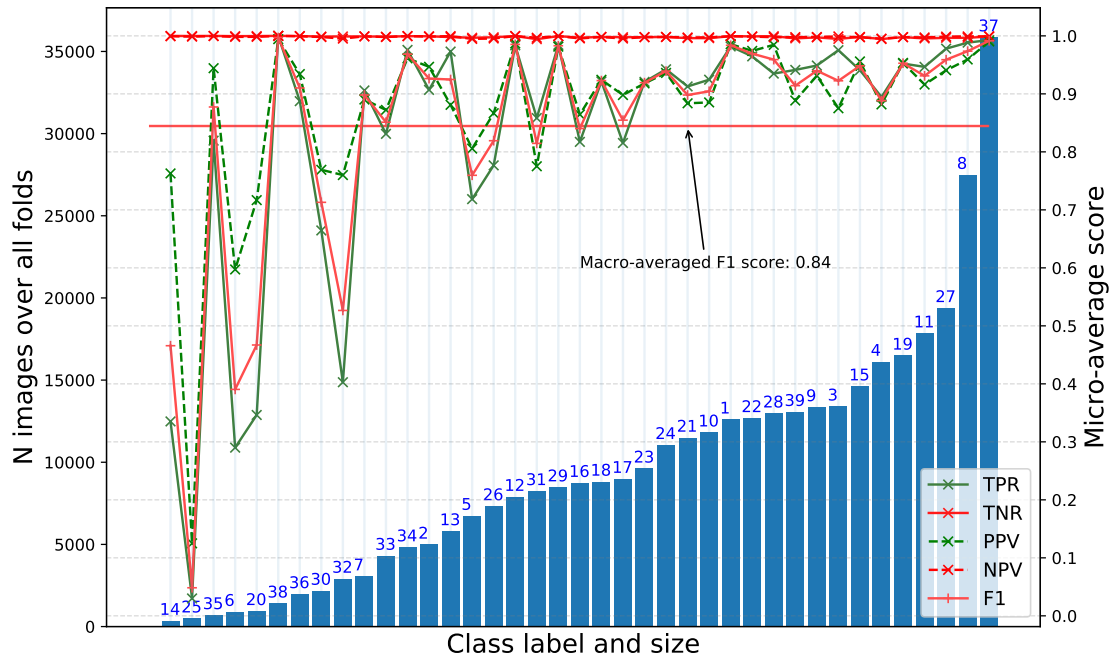


**Figure 5.2.** Reference model split 1 precision-recall curve. The large differences between class performance is well visible. Each curve corresponds to a one-vs-rest binary classification for each class separately

It has been argued, that the PR-curve is better at describing model performance in an imbalanced domain, than the return-on-characteristics (ROC), another popular threshold-based metric [73]. ROC curve calculates true positive rate against false positive rate over different thresholds. If the data is imbalanced, ROC curve might not necessarily reveal possible problems with the model, making it hard to distinguish between well-performing and poor models [73]. The next sections focus on more informative methods of visualizing model performance, focusing on displaying class-wise performance and their connection to the class size in an intuitive manner.

## 5.2 Imbalanced multi-class classification metrics and visualizations

To gain more insight on how a taxa classification model is performing, better metrics and visualizations are needed. The simplest way to visualize primary confusion matrix metrics is to plot them against the class sizes as seen in figure 5.3, called a *class-size-vs-score plot* from now on. The figure illustrates the positive performance metrics (TPR and PPV) as green and negative metrics (TNR, NPV) as red. Furthermore, the recall metrics



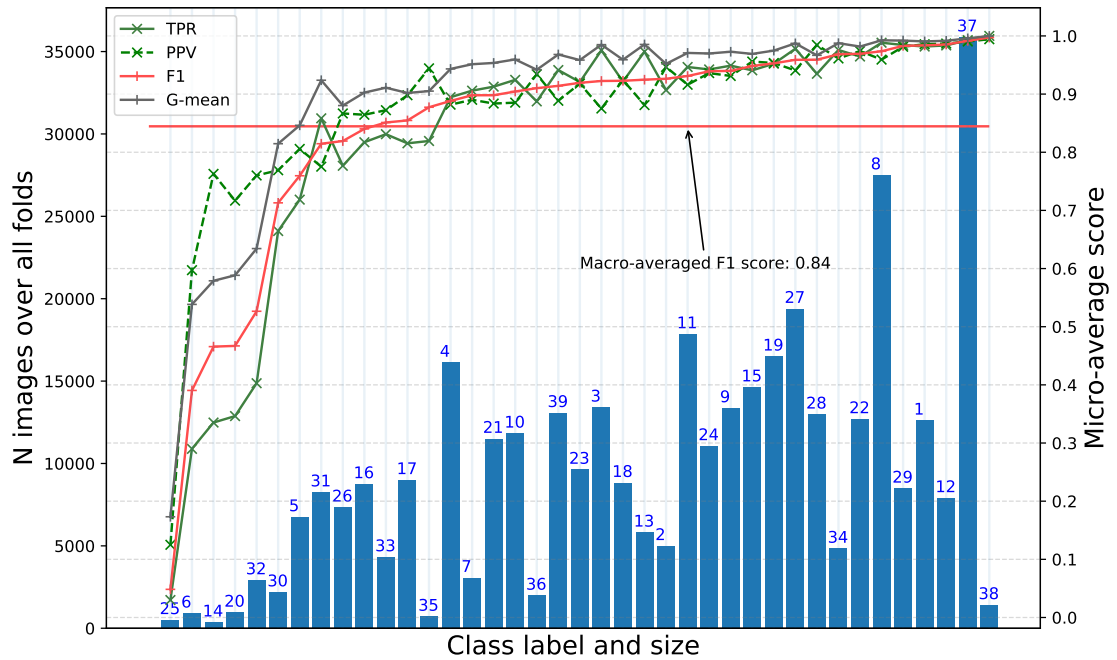
**Figure 5.3.** Reference model confusion matrix metrics with the F1-score highlighted. Macro-average of the whole model is marked with a horizontal line at 0.84. TNR and NPV values gain values close to 1, overlapping significantly in the figure.

(TPR, TNR) are marked with solid line and precision metrics (PPV, NPV) with dashed. Additionally, each plot visualizes the *macro*-averaged F1 score over classes. The figures presented here correspond to the *micro*-average of the cross-validated model over folds. According to Forman and Scholz [62], *micro*-averaging over the should provide a less biased output of each score.

It can be seen that as the class size increases, the scores increase and the variance among them decreases. Also, due to there being many classes and a large amount of true negatives for each class, the negative performance metrics (TNR, NPV) are close to 1. This makes it reasonable to focus on the positive performance metrics and F1 score which is derived from them. The *macro*-averaged F1 score is highlighted in orange in figure 5.3.

The definition of F1 as the harmonic mean of the two scores is highlighted when it is compared to the positive performance metrics. If either the precision or recall falls, it affects also the F1 score, but if both scores are fairly good, the F1 score is close to their arithmetic mean. Some insight can be gained by comparing class-wise scores to the *macro*-averaged mean, plotted as a horizontal line. It can be seen that there are several problems with using a single value as a metric, since majority of the classes have either low or high scores, but very few are actually close to the *macro*-averaged F1 score itself. Giving special attention to the classes gaining low F1 scores should be the first step in evaluating a multi-class classifier model.

Biologists should give special attention especially to the classes with low precision (PPV).



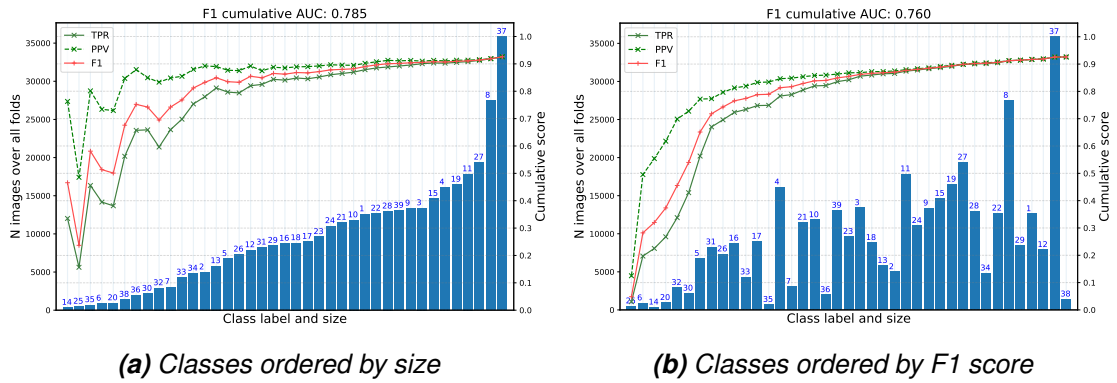
**Figure 5.4.** Reference model positive performance metrics, G-mean and F1 score ordered by F1 score. G-mean correlates strongly with TPR due to high class imbalance.

The precision score is analogous to the Bayesian probability, giving each class a "reliability score". Examining class-wise precision scores gives the model user an idea how probable is a certain classification. For example, if the reference model outputs a prediction *25-Nemoura cinerea*, the biologist should double check the sample since the model has only a probability of  $\sim 10\%$  being right. With classes like *1-Agapetus sp.* or *38-Sphaerium sp.* the samples are almost certainly classified correctly if they are detected and being proposed by the model.

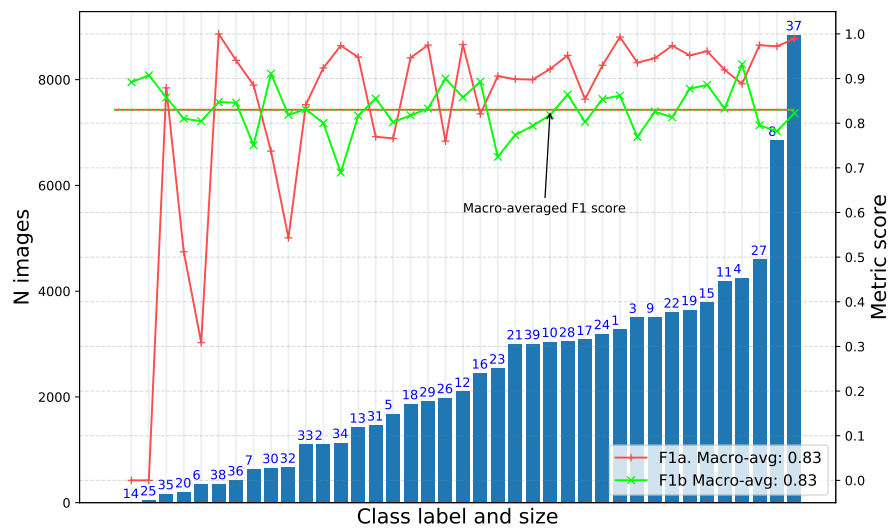
Another way of visualizing the class-wise scores is to order the scores in ascending order, as can be seen in figure 5.4. If it is desired, a threshold can be applied to choose classes with low performance to be the ones that are examined further by a biologist. In this case, it can be desired to end up with an model that outputs good scores for large classes to minimize the load on the examining biologist. Smaller classes with low reliability are easier to examine manually.

In addition to F1 and positive prediction scores, the G-mean score for all classes is visualized in the figure 5.4. It can be seen that due to the high class imbalance and large amount of classes, the specificity (TNR) values for all classes are close to 1, resulting in G-mean being approximately the TPR score squared, as explained in Section 3.3.3. Because of this it is more fruitful to focus on the TPR score, especially when evaluating taxa recognition models.

The class-size-vs-score plot visualizes class performance separately, giving an more overall view of the model performance. A single-score metric, for example the macro-average of the F1-score, can be the same for several very differently performing clas-



**Figure 5.5.** Reference model cumulative positive performance metrics and F1 score ordered in two ways. Cumulative metric score is calculated by taking the micro-average of all classes before the current class, ordered by some metric.



**Figure 5.6.** Example of two different classifiers with the same macro-averaged F1 score, but drastically different overall performance

sifiers. Figure 5.6 illustrates this with two example classifiers and their class-wise F1 scores. It can be seen, that the red example has both well and poorly performing classes, resulting in a F1 score of 0.83. The green classifier has a more uniform performance across classes, with less very well performing classes, but with the smaller classes performing as well as the rest. In many situations, the classifier of the green type is preferred over the red one; however, only using a single score to describe model performance makes it impossible to distinguish between these two models.

A novel alternative method of visualizing multi-class model performance is presented in the form of *cumulative metric plots*/ordered micro-average plots seen in figure 5.5. The *cumulative metric*/ordered micro-average of a class is defined to be the micro-average of the classes smaller or equal than it in a totally ordered set of classes. Take the ordered set of labels  $\mathcal{Y} = \{c_1, c_2, \dots, c_C\}$ , so that all classes have a relation  $c_i \leq c_{i+1}$  and that  $c_1 \leq c_i \leq c_C$  for all  $c_i \in \mathcal{Y}$  other than  $c_1$  and  $c_C$ . The order can be for example the class size, a specific metric or the relative importance of each class. Now, for a class  $c_A$ , the

cumulative metric is defined as

$$\text{cumulative}(c_A) = \frac{\sum_{i=1}^A TP_{c_i}}{\sum_{i=1}^A (TP_{c_i} + FP_{c_i})} \quad (5.7)$$

where  $A$  is the index of the class in the ordered set. Cumulative score is thus the same as micro-average, but calculated only for the current class and classes before it, ordered by an ordering function.

The cumulative metric plot visualizes how much each class contributes to the final, full-model micro-micro-average score presented in 5.3 that also the confusion matrix metrics converge to. With a non-cross-validated model the values converge to the regular micro-average. In an optimal situation, the plots in figure 5.5 would be horizontal lines, where each class would have similar F1 scores, and thus similar ratios between true positives and errors (FP/FN). However, the reference classifier performs poorly in classifying the smaller classes, resulting in a dispersed plot for both size-ordered and F1 ordered metrics.

An additional single-score metric can be derived from the cumulative metric plot ordered by the metric under inspection itself: the area under the cumulative metric curve. To reduce confusion to the traditional AUC score, this is marked with the notation  $AUC(\text{metric})_{\text{cumulative}}$ . For example, in figure 5.5b, the score of  $AUC(F1)_{\text{cumulative}} = 0.747$  is found by calculating the area under the increasingly monotonous cumulative F1 score curve, using a width of  $1/C$  for each class where  $C$  again is the number of classes. This single score brings to attention low scoring classes better than the macro-averages. On the other hand, the score does not take to account the relationship between the full-model score and the class scores. For example, the same score of  $AUC(PPV)_{\text{cumulative}} = 0.90$  is calculated if all classes have a the same precision of 0.9, or if 90% classes have a perfect precision, and 10% have 0.

### 5.3 Qualitative comparison

This chapter presented two different groups of performance evaluation: single-score metrics and class-wise metrics, former producing a single score describing the full model and the latter focusing on class performance. For single-score metrics, the different averaging methods were discussed, and classifier performance was evaluated using the classification error/accuracy, LCSE, precision, recall, and F1-scores. Additional single-score metrics can be derived from the precision-recall curve, and the cumulative micro-average curve. Average precision (AP) score can be calculated from the precision-recall curve over all classes and thresholds, describing general model performance well. The area under the cumulative micro-average curve describes also the model performance as a whole, penalizing poor overall performance.

For class-wise insight, two popular visualization techniques, confusion matrix and precision-recall -curves, were presented. Further study was done in form of class-size-vs-F1 -plots

ordered both by size and F1-score. It was seen that these methods illustrate the class imbalance problem and problematic class performance much better than single-score metrics. Additionally, a new method of visualizing class-wise performance was presented in the form of the cumulative score plot. Ordering the cumulative score in ascending order gave the possibility of calculating an cumulative AUC-score over classes, which can be used as an alternative single-score metric.

Quantitative comparison between different performance metrics is challenging due to different methods being useful in different situations. Table 5.4 summarizes the different metrics in a qualitative manner, listing their pros and cons from a taxonomic classification point of view.

**Table 5.4.** Comparison of different metrics discussed

<b>Metric</b>	<b>Pros</b>	<b>Cons</b>
Accuracy	Easy to use and understand	Can give misleading scores for poor classifiers
LCSE	Takes hierarchical classes to account	Similar problems as with accuracy
F1 score (macro)	Harmonic mean of precision and recall, both very suitable for imbalanced domains	Using general F-score with preference for either PPV or TPR could be more useful. Does not take to account performance distribution across classes
AP from precision-recall curve	Takes the precision-recall trade-off to account over different thresholds	Performs very similarly to basic F1-score but might be less intuitive. Knowledge over threshold performance has no practical use.
$AUC_{\text{cumulative}}$	Good score is gained with evenly good performance over all classes	Biased towards smaller classes (might be desired in some cases)
Confusion matrix	Visual and intuitive representation of between-class misclassifications	Hard to evaluate model performance
Precision-recall curves	Well-suitable for imbalanced data, visualizes the trade-off between precision and recall, and their effect on each class separately	Can be hard to interpret with large amount of classes with a separate curve for each class. One-vs-rest thresholds have no effect on practical use of the classifier
Class-size-vs-score-plot	General classifier performance can be easily evaluated. The effect of class size on performance is intuitively presented	Can become hard to interpret with a small or large amount of classes,
Cumulative score plot	Visualizes the contribution of each class to the total micro-average. Poor performance in several classes affects the plot cumulatively, making visual comparison of smaller differences easier	Uses non-intuitive cumulative micro-average as the main metric

## 6 COMPARISON OF LOSS-FUNCTIONS AND AUGMENTATION METHODS

The previous chapter showed how data imbalance affects the classification reliability of certain classes. It was seen that, at least with the benthic macroinvertebrate dataset, the class size was proportional to the performance measured with several metrics. In this section, attempts to improve the classification of the rarer classes are made. Several models were trained using different loss-functions, taking the imbalance to account with various methods. Additionally, two different weighting strategies are tested along with some of the loss-functions. Training a single model using the macroinvertebrate dataset takes an intense amount of computational resources; because of this, the compared models were validated using only the first train-test-split. For a better comparability, the cross-validated reference classifier is not used here, but instead the first test split classifier trained and tested with the same data as the comparison models.

This chapter presents three different loss functions designed for imbalanced, long-tailed datasets: focal loss, class-balanced loss and the weighted variant of the cross-entropy loss. Mathematical theory behind focal loss, class-balanced loss, and cross-entropy loss is described in Section 3.4, and in equations 3.35, 3.37 and 3.30 respectively.

This section uses three different ways to visualize classifier performance differences, all of which are modifications of the visualizations presented in Chapter 5. Classifier performance overview against the reference is visualized with the ascending class-performance plot similar to figure 5.4. Same performances in class-size-order similar to figure 5.3, modified to a "candlestick chart", are used to visualize absolute differences between class metrics. The third visualization method is the cumulative micro-average plot ordered by F1 score in figure 5.5b. Because the class orders change between models, the class sizes are not plotted to the same figure, due to the cumulative micro-average plot focusing on the *change* from the lowest F1-score classes to the highest scores. The cumulative comparison visualizes the possible improvements especially in the low-performing classes, and makes it possible to compare how large portion of classes fall below a certain performance threshold, regardless of the class size.

## 6.1 Categorical cross-entropy

The reference classifier explained in Section 4.2 uses the categorical cross-entropy loss as the optimization function. The reference model performance was compared to two different weighting schemes: normalized frequency, and inverse frequency.

Using normalized frequency, the number of samples in each class in the training set were calculated, and the value was normalized by the sum of all samples. This histogram value was then subtracted from 1 to produce lower weights for the larger classes. Because of the large amount of classes, the values are close to one for all classes, penalizing the larger classes only slightly.

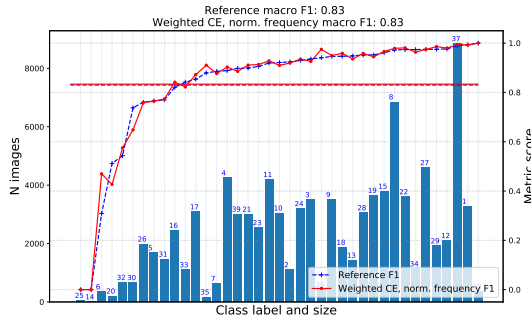
Inverse frequency, the more commonly used weighting method, penalizes the larger classes. The inverse frequency was calculated for each class by the inverse of the frequency, and normalizing this value with the sum of all inverse frequencies. Possible empty classes were replaced with the size of the second smallest class. This is done to ensure that the smallest weights do not reduce too much, which could happen if empty classes were given the weight 1. Inverse frequency gives larger classes values close to 0, penalizing them more than normalized frequency. In the practical situation, this can affect learning so that it could be more suitable to increase the learning rate. Due to comparison reasons, this was however not done in these experiments.

The overall class-wise performance of both weighting methods compared to the reference classifier can be seen in figure 6.1. For normalized frequency, it is possible to see, that because the weights are just slightly different from 1, the models are almost identical, with slight F1 score increases in the smallest classes. The inverse frequency however achieves significantly lower performance seen in all visualizations. This is probably due to the lower loss weights for most classes, which could be addressed by increasing the learning rate.

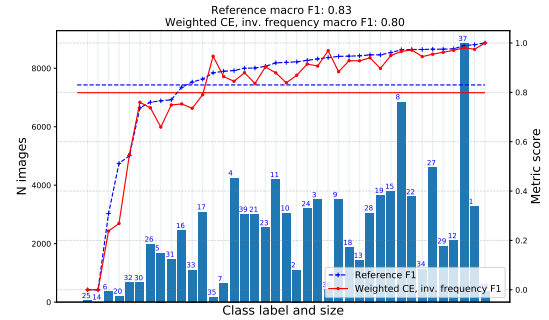
## 6.2 Focal loss

The focal loss function explained in Section 3.4.2 reduces the loss of well-classified examples, giving more weight to harder samples during training. In total, 5 different focal loss models were trained using gamma values:  $\gamma = \{1, 2, 4\}$  for non-weighted ( $\alpha = 1$ ) models, and  $\gamma = \{1, 2\}$  for models weighted by the normalized frequency. The weighted variants of the models were judged to be so similar to the reference models, that focus is here given to the  $\alpha = 1$  versions with different gamma values.

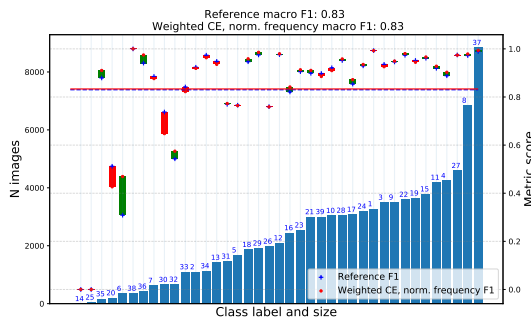
The comparison for  $\alpha = 1, \gamma = 2$  is shown in figure 6.2. Other gamma values can be seen in the appendices A.1, A.2, and A.3. It can be seen that using focal loss brings slight improvement to the smallest and previously poorly performing classes. However, the improvement is so slight that it can be said that this is probably due to noise. In the appendix figures, it is possible to see that changing the gamma value has little to no effect. Increasing gamma to  $\gamma = 4$  results actually in worse performance in poorly



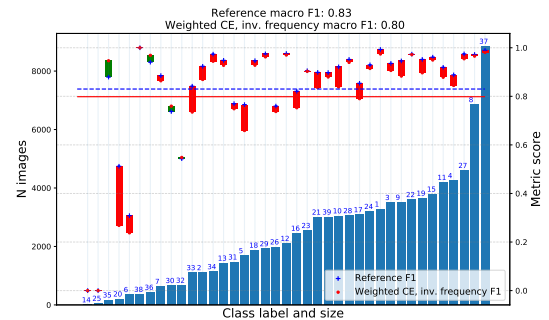
(a) Normalized frequency: ascending F1 plot



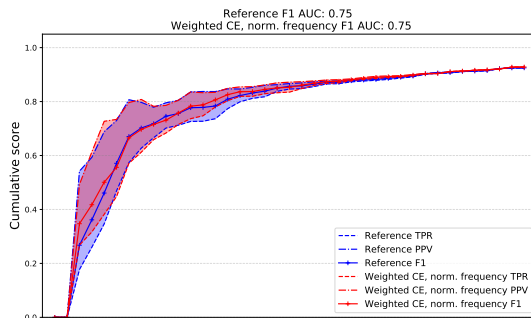
(b) Inverse frequency: ascending F1 plot



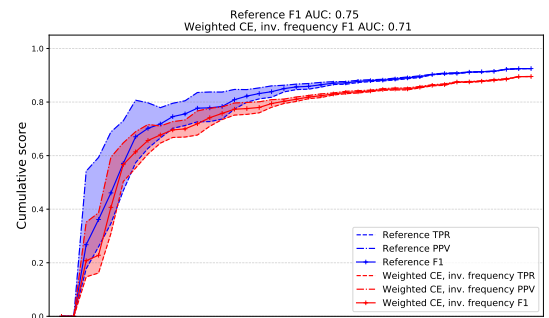
(c) Normalized frequency: class-wise changes in F1 scores



(d) Inverse frequency: class-wise changes in F1 scores

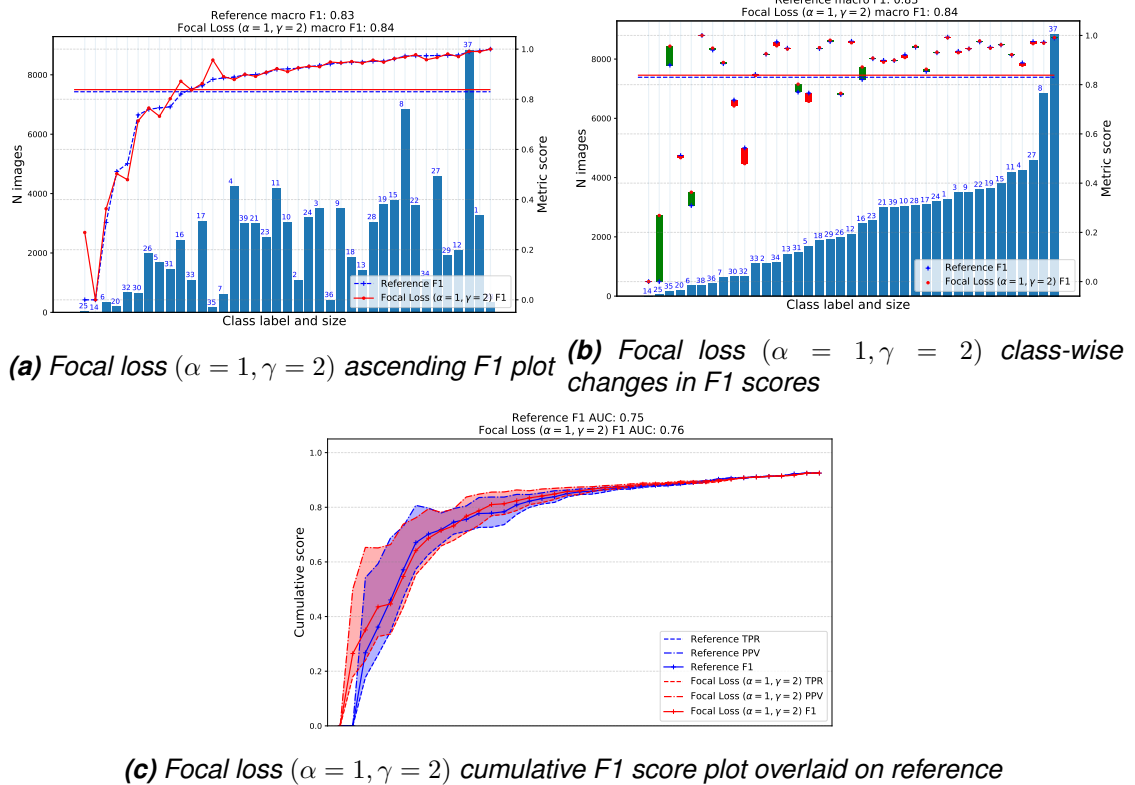


(e) Normalized frequency: Cumulative F1 score plot overlaid on reference



(f) Inverse frequency: Cumulative F1 score plot overlaid on reference

**Figure 6.1.** Weighted cross-entropy comparisons for normalized and inverse frequency weights. Differences are visualized with ascending F1 score, class-wise performance change "candlestick charts", and cumulative F1/precision/recall plots. Green "candle" indicates a performance improvement compared to the reference, and red a decrease in performance



**Figure 6.2.** Focal loss ( $\alpha = 1, \gamma = 2$ ) performance comparison

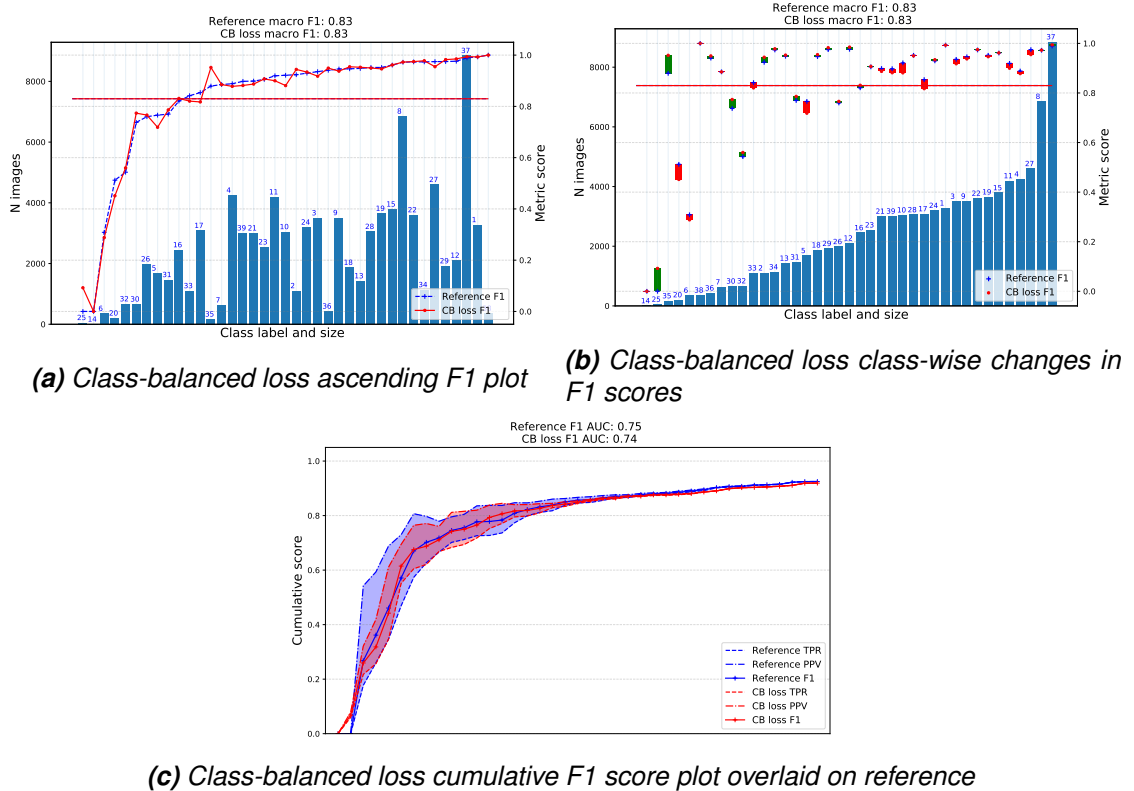
performing classes than smaller gamma values, although increasing gamma should give more attention to the hard examples.

The cumulative plot shows some improvements in low precision scores, resulting in a slightly higher F1 score across all classes. Recall however is still fairly poor, keeping the total performance on par with reference.

### 6.3 Class-balanced loss

Class-balanced loss attempts to solve the imbalance problems by weighing each class based on the available number of samples, and a hyperparameter  $\beta$ . The hyperparameter should be chosen according to total possible amount of samples  $N$ , which is fairly hard to estimate. In this experiment,  $N$  was chosen to be the number of samples in the maximum class, thus making  $\beta = (N - 1)/N \approx 0.999$ . The actual value changes between splits, but for split 1 present in the visualizations the actual value was  $\beta = 0.999967$ .

Figure 6.3 visualizes the class-balanced loss performance against the reference model. Again, the alternative loss function has little to no effect in improving the classification of the smallest classes. It seems, however, that focal loss has slight advantages over class-balanced loss, but most likely the differences are due to noise.



**Figure 6.3.** Class-balanced loss performance comparison

## 6.4 Resampling and data augmentation methods

Resampling and data augmentation are common methods used for imbalanced classification. The methods are generally regarded as good practice, when the dataset and use case allows for it [22, 23, 24]. For example, Provost [22] argues that often a problem of imbalanced classification is "solved" by just plain under- or oversampling. Provost also brings up the notion, that while other studies report that resampling can fix the unbalanced classification problem, other studies report the opposite.

In the quick experiments to test resampling as an improving method for this dataset, it was concluded that neither under- or oversampling was a good approach for the problem. However, both methods were very simple, and more sophisticated sampling methods could yield different results.

### 6.4.1 Under- and oversampling

Oversampling was performed so that the mean of all class sizes was calculated to produce a target  $a = \frac{1}{C} \sum_c N_c$  for oversampling. Here  $N_c$  is the number of samples in a class (in the training set),  $c$  is the class index, and  $C$  is the number of classes, 39 in this case. For each class, if  $N_c < a$ , class samples were duplicated *with augmentation* to reduce overfitting. Amount of duplicates for each class was the absolute difference  $|a - N_c|$ . If  $N_c > a$ , the class was kept untouched. The mean approach was used, to keep the

total amount of samples reasonable. Even with the mean approach, using training split 1, 92018 augmented duplicates were introduced to the training. This new dataset was then used for training the model similarly to the reference. The data augmentation performed for the duplicate samples is explained in Section 6.4.2.

For undersampling, similar target  $a$  was used, but in this case the classes with  $N_c > a$  were randomly pruned to contain only  $a$  samples. This resulted in 92026 fewer samples. The undersampling model was also trained similarly as the reference model.

A difference to the reference model was, that for both methods only 8 epochs were trained. This was due to both validation dataset accuracy scores were very poor with no improvement during training. Clear overfitting to the resampled datasets was noticed, as the training dataset accuracy was over 90% for both sampling methods. The training and validation accuracies at epoch 8 are displayed in table 6.1.

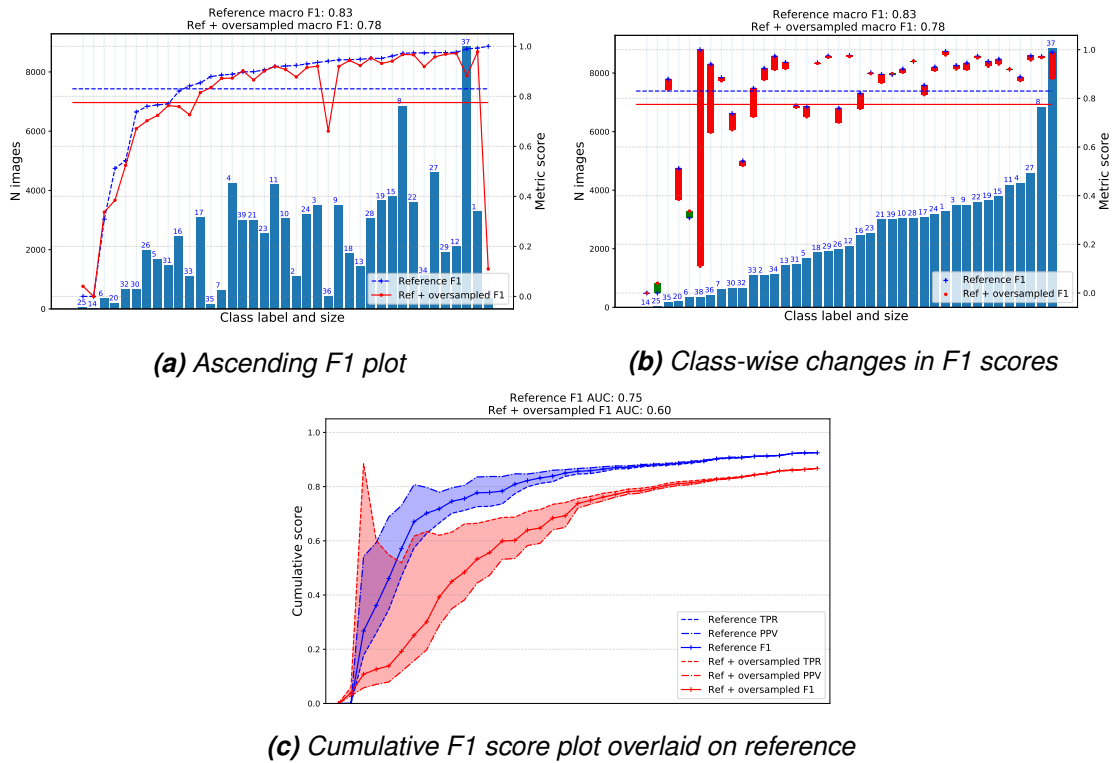
	Undersampling	Oversampling
Epochs	8	8
Training set acc (split1)	0.9217	0.9923
Validation set acc (split1)	0.0581	0.0968

**Table 6.1.** Performance of models trained using resampled datasets. Model clearly overfits the training data and fails to produce a comparable classifier

Oversampling was also tested by continuing the reference model training with the resampled dataset. During the 20 epochs of training, the convolutional layers were frozen, making the only trainable layers the final fully connected layers. The data distribution in the oversampled dataset seems to be so off, that the performance dropped drastically, as can be seen in figure 6.4.

It is possible to see, that because of the different data distribution, the classification performance decreases for several classes. The cumulative plot shows that the precision scores of each class are very low, making the classifier very unreliable. The comparison in figure 6.4 illustrates well the advantage of using the cumulative F1 score plot, compared to the metric comparisons for each class seen in 6.4a, or by judging performance based only on the macro-averaged F1 scores. Although the macro-averaged single-score metric difference can seem small, the cumulative plot reveals that the second model performance has a huge difference to the reference. One can compare these plots to the plots 6.1b and 6.1f, where the cross-entropy variations were compared, which show a similar difference between the averaged scores, but the cumulative plot shows that the overall performance is still fairly good with the "poorly performing" model.

From the tests performed here, it seems that resampling methods only reduce the performance of the classifiers. This is possibly due to the dataset being so large, that the ratio of duplicate samples and true samples grows substantially large. Because the model is a CNN, it is possible that also the smaller classes gain advantage of learning features from the more populous large classes. Reducing the amount of these samples reduces also



**Figure 6.4.** Reference model continued with oversampled data performance

the possibility of learning general features useful for all classes.

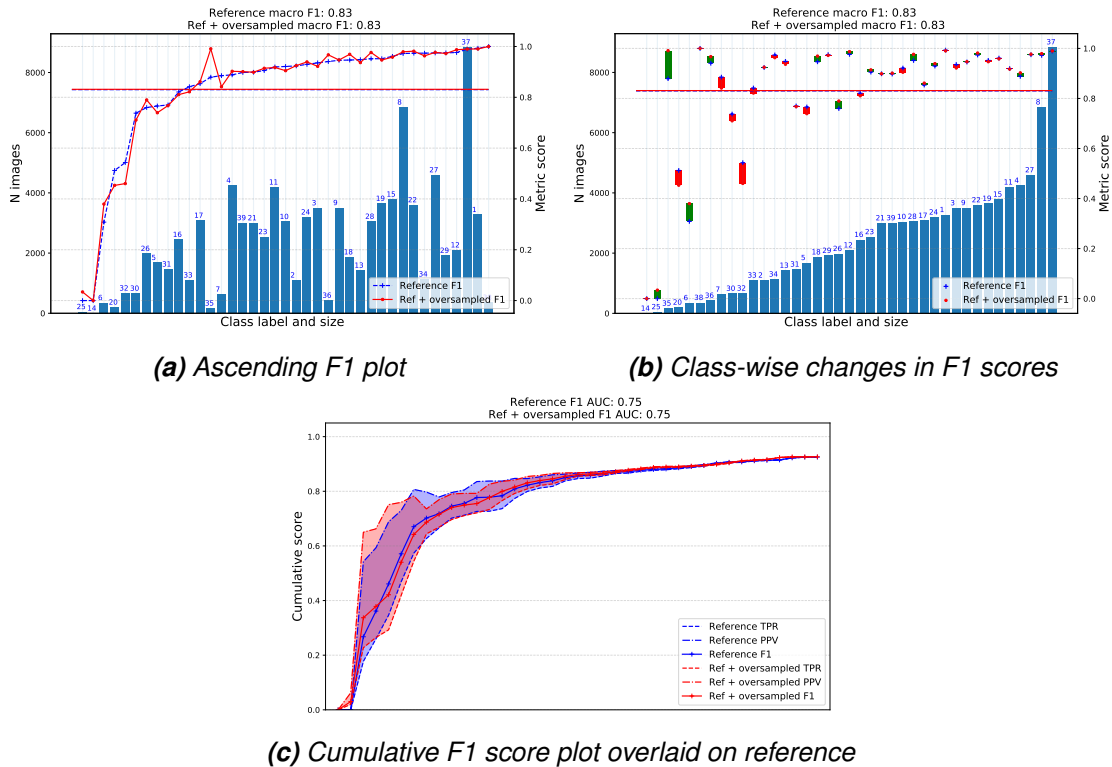
## 6.4.2 Data augmentation

To reduce the chance of overfitting, the duplicated data in the oversampled dataset was augmented by simple data-warping methods. In total four different methods were applied to the dataset: horizontal and vertical random flipping, brightness adjustment, and saturation adjustment. The augmentations had a uniform probability distribution among the possible choices, which was binary for the flipping operations, but continuous for brightness and saturation adjustments.

Augmentations were performed with TensorFlow's `tf.image`-library, which provides helpful functions for the operations. For brightness adjustment, function `tf.image.random_brightness`, with `max_delta`-parameter of `32/255` was used. For saturation, function `tf.image.random_saturation` with lower and upper bounds of `0.5` and `1.5` was used.

These augmentations were also tested for the non-resampled datasets. The full dataset was augmented using the methods described, and a model similar to the reference was trained. All other training parameters were the same, the only difference being that random augmentation functions were applied to each image, producing different augmentations for each learning epoch.

Using a fully augmented dataset produces a model that performs very similarly to the



**Figure 6.5.** Performance of a model trained with augmented data

reference. The comparisons can be seen in figure 6.5. It can be seen, that the differences are very small, but biggest improvements were achieved in the worst performing classes. Further research should be conducted whether this is consistent or due to noise. With a large dataset like this and with the current experiments, it seems that the augmentation most likely does not bring any more information to the training process.

## 6.5 Results

In order to combat problems presented by class imbalance, three common methods were tested on the benthic macroinvertebrate dataset: cost-sensitive training, resampling and data augmentation. It was seen, that most of these methods had little to no effect when compared to the "basic" training methods used on the reference classifier.

Three different cost-sensitive training methods were applied in form of loss-functions. Categorical cross-entropy (CE) loss was tested with two different weighting methods, inverse frequency weighting ( $wCE_{inv}$ ) and normalized frequency weighting ( $wCE_{norm}$ ), both of them performing similarly to the reference. It was noted that it would have been possible for inverse frequency to achieve better results: because of the weighting, the actual learning rate was almost an order of magnitude smaller than without weighting, which could have effected learning.

Focal loss (FL) and class-balanced loss (CB) were used as fairly new methods presented in recent years to be used especially in unbalanced domains. Focal loss was seen to

**Table 6.2.** Performance comparison of different models using single-score metrics. Best score of each column marked with **bold**

Model	acc	LCSE	F1 (macro)	AP	AUC <sub>cumulative</sub>
Reference (CE), split 1	0.9250	0.0458	0.8300	0.9723	0.7472
wCE <sub>inv</sub>	0.8954	0.0704	0.7984	0.9547	0.7067
wCE <sub>norm</sub>	<b>0.9281</b>	<b>0.0447</b>	0.8334	0.9727	0.7541
FL( $\gamma = 2, \alpha = 1$ )	0.9251	0.0460	<b>0.8385</b>	<b>0.9777</b>	<b>0.7599</b>
FL( $\gamma = 4, \alpha = 1$ )	0.9239	0.0460	0.8270	0.9770	0.7436
CB	0.9191	0.0503	0.8297	0.9673	0.7447
Oversample <sub>cont</sub>	0.8673	0.1053	0.7751	0.9278	0.5966
Augment	0.9264	0.0455	0.8314	0.9734	0.7492

have slight improvement in smaller classes, but it is entirely possible that this was due to random variation among training runs. It is however a result in itself that neither method was substantially worse or better, but actually very similar to plain cross-entropy.

Resampling methods provided very different results compared to loss-functions. Both over- and undersampled datasets failed to generalize to the validation data, heavily over-fitting to the training data. Continuing training the reference model with a resampled dataset (Oversample<sub>cont</sub>) failed to improve the smaller class performance, dropping the large class performance by a great deal instead. Just using augmented data without resampling (Augment) provided the most promising results, improving the scores of some of the smaller classes. Augmentation performance was in overall on a very similar level to the reference model.

The improvement test model performance against reference is summarized using single-score metrics in table 6.5. As stated before, single-score metrics have several problems, but are useful for quantitative comparison between similar classifiers. Several metrics are presented here, to highlight differences between classifiers, as well as the metrics themselves.

## 7 CONCLUSION

A notion that can be derived from the multi-class classification literature is that two things are often overlooked: the effect of unbalanced data on training, and the ambiguity of using a single-score metric for evaluation. Often model performance in machine learning papers is described using only accuracy, perhaps because it easily produces high, > 90% scores especially with imbalanced datasets populated with large, easy to classify classes. In a real-world situation, model robustness, reliability and uniform performance over all classes is more desired, and more attention should be given to metrics that take imbalances better to account. In many situations, it is also desired that large classes should not overshadow smaller ones. This is important especially in classifying benthic macroinvertebrates, where the smallest classes are often the most important ones.

To address these problems, two main areas of study were presented in this thesis: comparison of different performance metrics focusing on how well they take imbalance to account and attempting to improve these metrics with different methods. The main findings were:

1. Taxonomic classifier performance should be evaluated preferably on class-level, especially in imbalanced domains.
2. If a single score is used for classifier evaluation, it should take to account the overall performance over classes, without overshadowing the smaller classes.
3. Cost-sensitive learning, augmentation and rebalancing had little to no effect on improving the classification performance on the benthic macroinvertebrate dataset

Single score metrics were shown to have both ambiguity in how the score is calculated, as well as in how well the score describes overall performance. A good step forward is to at least understand the problems in using accuracy as a metric, but often the problem of unbalanced data is "fixed" in evaluation by using the F1 score, for example. Figure 5.6 showed, however, that two drastically differently performing classifiers can have the same macro-averaged F1-score. A better option, if it is possible, would be to compare models using intuitive visual representations.

Good single-score metrics for benthic macroinvertebrate classification that were highlighted were the plain macro-averaged F1-score and the area under the cumulative micro-average curve. The latter highlights uniform performance over all classes, when the former is a common and easy to use metric, tolerable for simplified comparison.

Good visualizations for imbalanced classification of benthic macroinvertebrates turned out to be the precision-recall curve and class-size-vs-score plot. The former is commonly used, and highlights problems well also in imbalanced domains, but is fairly hard to read especially with multiple classes. The class-size-vs-score plot makes it easy to evaluate the reliability of each class separately. This could provide to be important for biologists possibly evaluating whether they should trust the output of a classifier or not.

The improvement methods used for assessing the imbalance problem would need further study. Current experiments did not yield better or worse results compared to using plain cross-entropy as a loss-function, without rebalancing or augmentation. The classifier itself was fairly powerful in learning the representations, so it could be possible that rebalancing schemes and loss functions could have more effect used with simpler architectures than the one used here.

Overall, the problem of imbalanced data is an important and unfortunately fairly overlooked problem in automated biomonitoring systems, and more research would be needed in this area. Biologists and environmental scientists should be involved in the research, especially by providing insight on the weighting and importance of different classes. Hierarchical nature of the data, characteristic to taxonomic classification problems, was overlooked in this thesis for simplicity. More study in this direction could, however, provide important insight in better metrics in the taxonomic classification domain.

## REFERENCES

- [1] Gleick, P. H. Water and conflict: Fresh water resources and international security. *International security* 18.1 (1993), 79–112.
- [2] ECOSOC, U. Special Edition: Progress towards the Sustainable Development Goals Report of the Secretary-General. *Advanced unedited version. New York (US): United Nations* (2019).
- [3] Barbour, M. T., Gerritsen, J., Snyder, B. D., Stribling, J. B. et al. *Rapid bioassessment protocols for use in streams and wadeable rivers: periphyton, benthic macroinvertebrates and fish*. Vol. 339. US Environmental Protection Agency, Office of Water Washington, DC, 1999.
- [4] Ärje, J., Raitoharju, J., Iosifidis, A., Tirronen, V., Meissner, K., Gabbouj, M., Kiranyaz, S. and Kärkkäinen, S. Human experts vs. machines in taxa recognition. *arXiv preprint arXiv:1708.06899* (2017).
- [5] Kalafi, E. Y., Town, C. and Dhillon, S. K. How automated image analysis techniques help scientists in species identification and classification?: *Folia morphologica* 77.2 (2018), 179–193.
- [6] Joutsijoki, H., Meissner, K., Gabbouj, M., Kiranyaz, S., Raitoharju, J., Ärje, J., Kärkkäinen, S., Tirronen, V., Turpeinen, T. and Juhola, M. Evaluating the performance of artificial neural networks for the classification of freshwater benthic macroinvertebrates. *Ecological informatics* 20 (2014), 1–12.
- [7] Kiranyaz, S., Ince, T., Pulkkinen, J., Gabbouj, M., Ärje, J., Kärkkäinen, S., Tirronen, V., Juhola, M., Turpeinen, T. and Meissner, K. Classification and retrieval on macroinvertebrate image databases. *Computers in biology and medicine* 41.7 (2011), 463–472.
- [8] Raitoharju, J., Riabchenko, E., Ahmad, I., Iosifidis, A., Gabbouj, M., Kiranyaz, S., Tirronen, V., Ärje, J., Kärkkäinen, S. and Meissner, K. Benchmark database for fine-grained image classification of benthic macroinvertebrates. *Image and Vision Computing* 78 (2018), 73–83.
- [9] Culverhouse, P. F., Simpson, R., Ellis, R., Lindley, J., Williams, R., Parisini, T., Reguera, B., Bravo, I., Zoppoli, R., Earnshaw, G. et al. Automatic classification of field-collected dinoflagellates by artificial neural network. *Marine Ecology Progress Series* 139 (1996), 281–287.
- [10] O'Neill, M., Gauld, I., Gaston, K. and Weeks, P. Daisy: an automated invertebrate identification system using holistic vision techniques. *Proceedings of the Inaugural Meeting BioNET-INTERNATIONAL Group for Computer-Aided Taxonomy (BIG-CAT)*. 2000, 13–22.

- [11] Wei, X.-S., Xie, C.-W., Wu, J. and Shen, C. Mask-CNN: Localizing parts and selecting descriptors for fine-grained bird species categorization. *Pattern Recognition* 76 (2018), 704–714.
- [12] Van Horn, G., Mac Aodha, O., Song, Y., Cui, Y., Sun, C., Shepard, A., Adam, H., Perona, P. and Belongie, S. The inaturalist species classification and detection dataset. *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2018, 8769–8778.
- [13] Cui, Y., Song, Y., Sun, C., Howard, A. and Belongie, S. Large scale fine-grained categorization and domain-specific transfer learning. *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2018, 4109–4118.
- [14] Cui, Y., Jia, M., Lin, T.-Y., Song, Y. and Belongie, S. Class-balanced loss based on effective number of samples. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2019, 9268–9277.
- [15] Goutte, C. and Gaussier, E. A probabilistic interpretation of precision, recall and F-score, with implication for evaluation. *European Conference on Information Retrieval*. Springer. 2005, 345–359.
- [16] Bekkar, M., Djemaa, H. K. and Alitouche, T. A. Evaluation measures for models assessment over imbalanced data sets. *J Inf Eng Appl* 3.10 (2013).
- [17] Hossin, M. and Sulaiman, M. A review on evaluation metrics for data classification evaluations. *International Journal of Data Mining & Knowledge Management Process* 5.2 (2015), 1.
- [18] Powers, D. M. Evaluation: from precision, recall and F-measure to ROC, informedness, markedness and correlation. (2011).
- [19] Sokolova, M. and Lapalme, G. A systematic analysis of performance measures for classification tasks. *Information processing & management* 45.4 (2009), 427–437.
- [20] Japkowicz, N. and Stephen, S. The class imbalance problem: A systematic study. *Intelligent data analysis* 6.5 (2002), 429–449.
- [21] Krawczyk, B. Learning from imbalanced data: open challenges and future directions. *Progress in Artificial Intelligence* 5.4 (2016), 221–232.
- [22] Provost, F. Machine learning from imbalanced data sets 101. *Proceedings of the AAAI'2000 workshop on imbalanced data sets*. Vol. 68. 2000. AAAI Press. 2000, 1–3.
- [23] Huang, C., Li, Y., Change Loy, C. and Tang, X. Learning deep representation for imbalanced classification. *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2016, 5375–5384.
- [24] Wang, J. and Perez, L. The effectiveness of data augmentation in image classification using deep learning. *Convolutional Neural Networks Vis. Recognit* (2017), 11.
- [25] Liu, A., Ghosh, J. and Martin, C. E. Generative Oversampling for Mining Imbalanced Datasets. *DMIN*. 2007, 66–72.

- [26] Mikołajczyk, A. and Grochowski, M. Data augmentation for improving deep learning in image classification problem. *2018 international interdisciplinary PhD workshop (IIPhDW)*. IEEE. 2018, 117–122.
- [27] Batista, G. E., Prati, R. C. and Monard, M. C. A study of the behavior of several methods for balancing machine learning training data. *ACM SIGKDD explorations newsletter* 6.1 (2004), 20–29.
- [28] Drummond, C., Holte, R. C. et al. C4. 5, class imbalance, and cost sensitivity: why under-sampling beats over-sampling. *Workshop on learning from imbalanced datasets II*. Vol. 11. Citeseer. 2003, 1–8.
- [29] Chawla, N. V., Bowyer, K. W., Hall, L. O. and Kegelmeyer, W. P. SMOTE: synthetic minority over-sampling technique. *Journal of artificial intelligence research* 16 (2002), 321–357.
- [30] McCarthy, K., Zabar, B. and Weiss, G. Does cost-sensitive learning beat sampling for classifying rare classes?: *Proceedings of the 1st international workshop on Utility-based data mining*. 2005, 69–77.
- [31] Elkan, C. The foundations of cost-sensitive learning. *International joint conference on artificial intelligence*. Vol. 17. 1. Lawrence Erlbaum Associates Ltd. 2001, 973–978.
- [32] Lin, T.-Y., Goyal, P., Girshick, R., He, K. and Dollár, P. Focal loss for dense object detection. *Proceedings of the IEEE international conference on computer vision*. 2017, 2980–2988.
- [33] Cairns, J. and Pratt, J. R. A history of biological monitoring using benthic macroinvertebrates. *Freshwater biomonitoring and benthic macroinvertebrates* 10 (1993), 27.
- [34] Gerhardt, A. Bioindicator species and their use in biomonitoring. *Environmental monitoring* 1 (2002), 77–123.
- [35] Hassall, A. H. *A microscopic examination of the water supplied to the inhabitants of London and the suburban districts: illustrated by coloured plates exhibiting the living animal and vegetable productions in Thames and other waters, as supplied by several companies: with an examination, microscopic and general, of their sources of supply as well as of the Henley-on-Thames and Watford plans, etc.* Samuel Highley, 1850.
- [36] Moog, O., Schmutz, S. and Schwarzingler, I. Biomonitoring and bioassessment. *Riverine Ecosystem Management* (2018), 371.
- [37] Hershey, A. E., Lamberti, G. A., Chaloner, D. T. and Northington, R. M. Aquatic insect ecology. *Ecology and classification of North American freshwater invertebrates*. Elsevier, 2010, 659–694.
- [38] Ylikörkkö, J., Christensen, G. N., Andersen, H. J., Denisov, D., Amundsen, P.-A., Terentjev, P. and Jelkänen, E. Environmental Monitoring Programme for Aquatic Ecosystems in the Norwegian, Finnish and Russian Border Area; Updated Implementation Guidelines. (2015).

- [39] Postel, S. L., Daily, G. C. and Ehrlich, P. R. Human appropriation of renewable fresh water. *Science* 271.5250 (1996), 785–788.
- [40] Lindenmayer, D. B., Margules, C. R. and Botkin, D. B. Indicators of biodiversity for ecologically sustainable forest management. *Conservation biology* 14.4 (2000), 941–950.
- [41] Johnson, R. K., Wiederholm, T., Rosenberg, D. M. et al. Freshwater biomonitoring using individual organisms, populations, and species assemblages of benthic macroinvertebrates. *Freshwater biomonitoring and benthic macroinvertebrates* (1993), 40–158.
- [42] Mulgrew, A., Williams, P. et al. *Biomonitoring of air quality using plants*. WHO Collaborating Centre for Air Quality Management and Air Pollution Control, 2000.
- [43] Landres, P. B., Verner, J. and Thomas, J. W. Ecological uses of vertebrate indicator species: a critique. *Conservation biology* 2.4 (1988), 316–328.
- [44] Siddig, A. A., Ellison, A. M., Ochs, A., Villar-Leeman, C. and Lau, M. K. How do ecologists select and use indicator species to monitor ecological change? Insights from 14 years of publication in Ecological Indicators. *Ecological Indicators* 60 (2016), 223–230.
- [45] Keck, F., Vasselon, V., Tapolczai, K., Rimet, F. and Bouchez, A. Freshwater biomonitoring in the Information Age. *Frontiers in Ecology and the Environment* 15.5 (2017), 266–274.
- [46] Warren Jr, M. L. and Burr, B. M. Status of freshwater fishes of the United States: overview of an imperiled fauna. *Fisheries* 19.1 (1994), 6–18.
- [47] Brown, L. R. and May, J. T. *Benthic macroinvertebrate assemblages and their relations with environmental variables in the Sacramento and San Joaquin River drainages, California, 1993-1997*. 4125. US Department of the Interior, US Geological Survey, 2000.
- [48] Gaston, K. J. and O'Neill, M. A. Automated species identification: why not?: *Philosophical Transactions of the Royal Society of London. Series B: Biological Sciences* 359.1444 (2004), 655–667.
- [49] Bishop, C. M. *Pattern recognition and machine learning*. Springer, 2006.
- [50] Goodfellow, I., Bengio, Y. and Courville, A. *Deep Learning*. <http://www.deeplearningbook.org>. MIT press, 2016.
- [51] Russell, S. J. and Norvig, P. *Artificial intelligence: a modern approach*. Malaysia; Pearson Education Limited, 2016.
- [52] Shalev-Shwartz, S. and Ben-David, S. *Understanding machine learning: From theory to algorithms*. Cambridge university press, 2014.
- [53] Boyd, S. and Vandenberghe, L. *Convex optimization*. Cambridge university press, 2004.
- [54] Nocedal, J. and Wright, S. *Numerical optimization*. Springer Science & Business Media, 2006.
- [55] Hastie, T., Tibshirani, R. and Friedman, J. *The elements of statistical learning: data mining, inference, and prediction*. Springer Science & Business Media, 2009.

- [56] Murphy, K. P. *Machine learning: a probabilistic perspective*. MIT press, 2012.
- [57] Friedman, J. H. On bias, variance, 0/1—loss, and the curse-of-dimensionality. *Data mining and knowledge discovery* 1.1 (1997), 55–77.
- [58] Shin, H.-C., Roth, H. R., Gao, M., Lu, L., Xu, Z., Nogues, I., Yao, J., Mollura, D. and Summers, R. M. Deep convolutional neural networks for computer-aided detection: CNN architectures, dataset characteristics and transfer learning. *IEEE transactions on medical imaging* 35.5 (2016), 1285–1298.
- [59] Nguyen, T. and Sanner, S. Algorithms for direct 0–1 loss optimization in binary classification. *International Conference on Machine Learning*. 2013, 1085–1093.
- [60] Verma, N., Mahajan, D., Sellamanickam, S. and Nair, V. Learning hierarchical similarity metrics. *2012 IEEE conference on computer vision and pattern recognition*. IEEE. 2012, 2280–2287.
- [61] Van Rijsbergen, C. J. *Information Retrieval. 2nd. Newton, MA*. <http://www.dcs.gla.ac.uk/Keith/Preface.html>. Accessed: 2020-01-22. 1979.
- [62] Forman, G. and Scholz, M. Apples-to-apples in cross-validation studies: pitfalls in classifier performance measurement. *Acm Sigkdd Explorations Newsletter* 12.1 (2010), 49–57.
- [63] Fleming, P. J. and Wallace, J. J. How not to lie with statistics: the correct way to summarize benchmark results. *Communications of the ACM* 29.3 (1986), 218–221.
- [64] Hand, D. and Christen, P. A note on using the F-measure for evaluating record linkage algorithms. *Statistics and Computing* 28.3 (2018), 539–547.
- [65] Branco, P., Torgo, L. and Ribeiro, R. P. A survey of predictive modeling on imbalanced domains. *ACM Computing Surveys (CSUR)* 49.2 (2016), 1–50.
- [66] Shannon, C. E. A mathematical theory of communication. *Bell system technical journal* 27.3 (1948), 379–423.
- [67] Simard, P. Y., Steinkraus, D., Platt, J. C. et al. Best practices for convolutional neural networks applied to visual document analysis. *Icdar*. Vol. 3. 2003. 2003.
- [68] Wong, S. C., Gatt, A., Stamatescu, V. and McDonnell, M. D. Understanding data augmentation for classification: when to warp?: *2016 international conference on digital image computing: techniques and applications (DICTA)*. IEEE. 2016, 1–6.
- [69] Szegedy, C., Vanhoucke, V., Ioffe, S., Shlens, J. and Wojna, Z. Rethinking the inception architecture for computer vision. *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2016, 2818–2826.
- [70] Deng, J., Dong, W., Socher, R., Li, L.-J., Li, K. and Fei-Fei, L. ImageNet: A Large-Scale Hierarchical Image Database. *CVPR09*. 2009.
- [71] Abadi, M., Barham, P., Chen, J., Chen, Z., Davis, A., Dean, J., Devin, M., Ghemawat, S., Irving, G., Isard, M. et al. Tensorflow: A system for large-scale machine learning. *12th {USENIX} Symposium on Operating Systems Design and Implementation ({OSDI} 16)*. 2016, 265–283.
- [72] Kingma, D. P. and Ba, J. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980* (2014).

- [73] Saito, T. and Rehmsmeier, M. The precision-recall plot is more informative than the ROC plot when evaluating binary classifiers on imbalanced datasets. *PloS one* 10.3 (2015).

## A APPENDIX

**Table A.1.** Taxa with their taxonomic classifications and class sizes

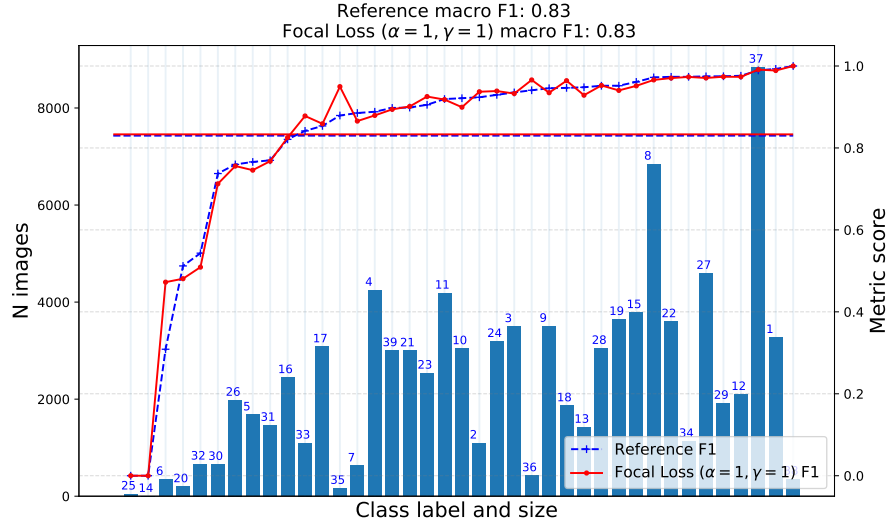
Taxa	Species	Genus	Family	Order	#specimens	#images
8-Elmis aenea	Elmis aenea	Elmis	Elmidae	Coleoptera	648	32398
22-Limnius volckmari	Limnius volckmari	Limnius	Elmidae	Coleoptera	314	15621
28-Oulimnius tuberculatus	Oulimnius tuberculatus	Oulimnius	Elmidae	Coleoptera	335	16674
12-Hydraena sp.	-	Hydraena	Hydraenidae	Coleoptera	198	9900
37-Simuliidae	-	-	Simuliidae	Diptera	887	44240
2-Ameletus inopinatus	Ameletus inopinatus	Ameletus	Ameletidae	Ephemeroptera	127	6346
4-Baetis rhodani	Baetis rhodani	Baetis	Baetidae	Ephemeroptera	404	19829
5-Baetis vernus group	Baetis vernus	Baetis	Baetidae	Ephemeroptera	176	8588
9-Ephemerella aurivillii	Ephemerella aurivillii	Ephemerella	Ephemerellidae	Ephemeroptera	356	16458
10-Ephemerella mucronata	Ephemerella mucronata	Ephemerella	Ephemerellidae	Ephemeroptera	304	15175
11-Heptagenia sulphurea	Heptagenia sulphurea	Heptagenia	Heptageniidae	Ephemeroptera	438	21502
17-Kageronia fuscogrisea	Kageronia fuscogrisea	Kageronia	Heptageniidae	Ephemeroptera	222	10826
19-Leptophlebia sp.	-	Leptophlebia	Leptophlebiidae	Ephemeroptera	412	20366
35-Sialis sp.	-	Sialis	Sialidae	Megaloptera	26	1162
6-Capnopsis schilleri	Capnopsis schilleri	Capnopsis	Capniidae	Plecoptera	21	1050
20-Leuctra nigra	Leuctra nigra	Leuctra	Leuctridae	Plecoptera	27	1350
21-Leuctra sp.	-	Leuctra	Leuctridae	Plecoptera	298	14899
3-Amphinemura borealis	Amphinemura borealis	Amphinemura	Nemouridae	Plecoptera	322	16100
25-Nemoura cinerea	Nemoura cinerea	Nemoura	Nemouridae	Plecoptera	16	800
26-Nemoura sp.	-	Nemoura	Nemouridae	Plecoptera	187	9314
33-Protonemura sp.	-	Protonemura	Nemouridae	Plecoptera	100	4908
7-Diura sp.	-	Diura	Perlodiae	Plecoptera	98	4427
16-Isoperla sp.	-	Isoperla	Perlodiae	Plecoptera	243	12148
39-Taeniopteryx nebulosa	Taeniopteryx nebulosa	Taeniopteryx	Taeniopterygidae	Plecoptera	331	16325
23-Micrasema gelidum	Micrasema gelidum	Micrasema	Brachycentridae	Trichoptera	233	11528
24-Micrasema setiferum	Micrasema setiferum	Micrasema	Brachycentridae	Trichoptera	323	13819
1-Agapetus sp.	-	Agapetus	Glossosomatidae	Trichoptera	290	14387
36-Silo pallipes	Silo pallipes	Silo	Goeridae	Trichoptera	56	2658
13-Hydropsyche pellucidula	Hydropsyche pellucidula	Hydropsyche	Hydropsychidae	Trichoptera	192	6513
14-Hydropsyche saxonica	Hydropsyche saxonica	Hydropsyche	Hydropsychidae	Trichoptera	17	490
15-Hydropsyche siltalai	Hydropsyche siltalai	Hydropsyche	Hydropsychidae	Trichoptera	395	19456
29-Oxyethira sp.	-	Oxyethira	Hydroptiliidae	Trichoptera	218	10381
18-Lepidostoma hirtum	Lepidostoma hirtum	Lepidostoma	Lepidostomatidae	Trichoptera	267	10982
27-Neureclipsis bimaculata	Neureclipsis bimaculata	Neureclipsis	Polycentropodidae	Trichoptera	477	23721
30-Plectrocnemia sp.	-	Plectrocnemia	Polycentropodidae	Trichoptera	63	3015
31-Polycentropus flavomaculatus	Polycentropus flavomaculatus	Polycentropus	Polycentropodidae	Trichoptera	224	11005
32-Polycentropus irroratus	Polycentropus irroratus	Polycentropus	Polycentropodidae	Trichoptera	59	2917
36-Rhyacophila nubila	Rhyacophila nubila	Rhyacophila	Rhyacophiliidae	Trichoptera	177	6993
38-Sphaerium sp.	-	Sphaerium	Sphaeridae	Veneroida	150	1733
					<b>9631</b>	<b>460004</b>

**Table A.2.** Reference model confusion matrix values summed over folds

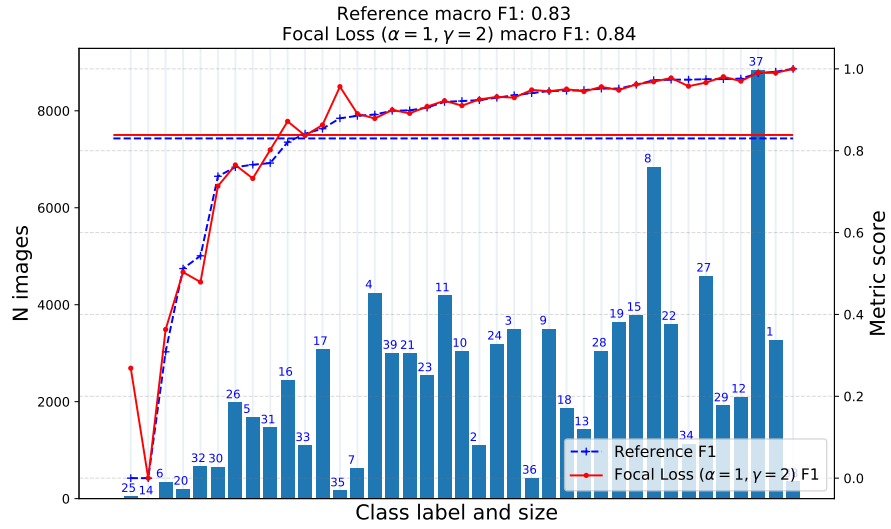
<b>c</b>	<b>TP</b>	<b>FP</b>	<b>TN</b>	<b>FN</b>	<b>Positive</b>	<b>Negative</b>
1	12377	357333	179	230	12607	357512
2	4527	364872	255	465	4992	365127
3	13069	354855	1864	331	13400	356719
4	14422	352066	1933	1698	16120	353999
5	4847	362204	1169	1899	6746	363373
6	261	369043	176	639	900	369219
7	2773	366716	342	288	3061	367058
8	27176	341470	1153	320	27496	342623
9	12676	355823	937	683	13359	356760
10	10926	356884	1416	893	11819	358300
11	16894	350723	1548	954	17848	352271
12	7771	362116	103	129	7900	362219
13	5654	363545	763	157	5811	364308
14	119	369727	37	236	355	369764
15	13758	354862	634	865	14623	355496
16	7149	360250	1121	1599	8748	361371
17	7334	360291	833	1661	8995	361124
18	8131	360629	666	693	8824	361295
19	15707	352857	767	788	16495	353624
20	329	369039	130	621	950	369169
21	10497	357238	1381	1003	11500	358619
22	12243	357101	326	449	12692	357427
23	8860	359705	791	763	9623	360496
24	10418	358353	711	637	11055	359064
25	15	369514	105	485	500	369619
26	5695	361911	876	1637	7332	362787
27	18960	349535	1197	427	19387	350732
28	12131	356953	191	844	12975	357144
29	8347	361482	158	132	8479	361640
30	1452	367498	436	733	2185	367934
31	7082	359814	2054	1169	8251	361868
32	1175	366831	371	1742	2917	367202
33	3586	365279	525	729	4315	365804
34	4713	365099	190	117	4830	365289
35	576	369382	34	127	703	369416
36	1758	368014	124	223	1981	368138
37	35607	333906	346	260	35867	334252
38	1410	368701	8	0	1410	368709
39	12301	355508	1543	767	13068	357051

**Table A.3.** Reference model confusion matrix values for test split 1

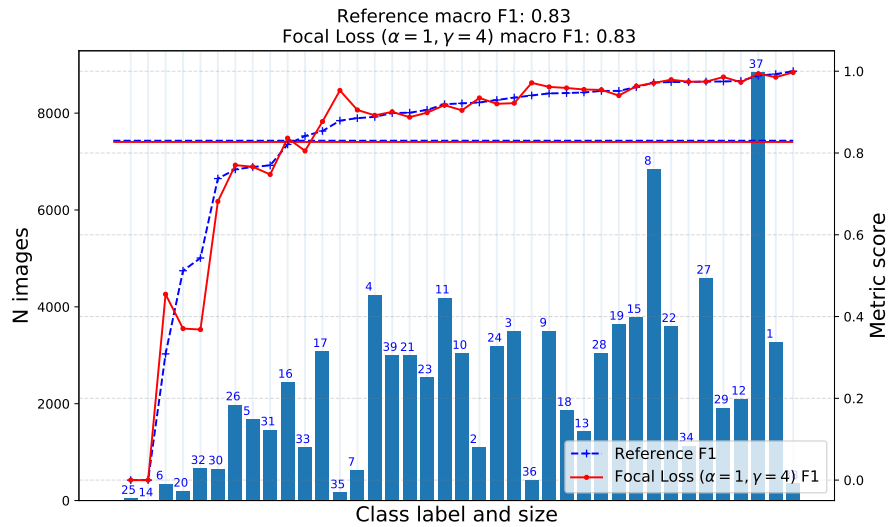
<b>c</b>	<b>TP</b>	<b>FP</b>	<b>TN</b>	<b>FN</b>	<b>Positive</b>	<b>Negative</b>
1	3256	27	89383	19	3275	89410
2	992	56	91529	108	1100	91585
3	3402	371	88814	98	3500	89185
4	3827	541	87896	421	4248	88437
5	1264	356	90648	417	1681	91004
6	71	39	92296	279	350	92335
7	594	119	91937	35	629	92056
8	6771	310	85527	77	6848	85837
9	3344	222	88957	162	3506	89179
10	2847	289	89352	197	3044	89641
11	4053	575	87921	136	4189	88496
12	2031	30	90555	69	2100	90585
13	1370	95	91165	55	1425	91260
14	0	7	92678	0	0	92685
15	3642	148	88749	146	3788	88897
16	1971	383	89854	477	2448	90237
17	2387	122	89480	696	3083	89602
18	1791	125	90692	77	1868	90817
19	3481	186	88852	166	3647	89038
20	85	47	92438	115	200	92485
21	2707	318	89367	293	3000	89685
22	3448	35	89050	152	3600	89085
23	2229	154	89990	312	2541	90144
24	3004	266	89226	189	3193	89492
25	0	14	92621	50	50	92635
26	1338	202	90501	644	1982	90703
27	4563	204	87885	33	4596	88089
28	2864	104	89531	186	3050	89635
29	1865	42	90724	54	1919	90766
30	465	136	91889	195	660	92025
31	1192	440	90780	273	1465	91220
32	283	92	91926	384	667	92018
33	885	118	91467	215	1100	91585
34	1109	35	91516	25	1134	91551
35	131	0	92518	36	167	92518
36	413	38	92220	14	427	92258
37	8823	168	83672	22	8845	83840
38	356	0	92329	0	356	92329
39	2882	535	89146	122	3004	89681



(a) Ascending F1 with  $\gamma = 1$

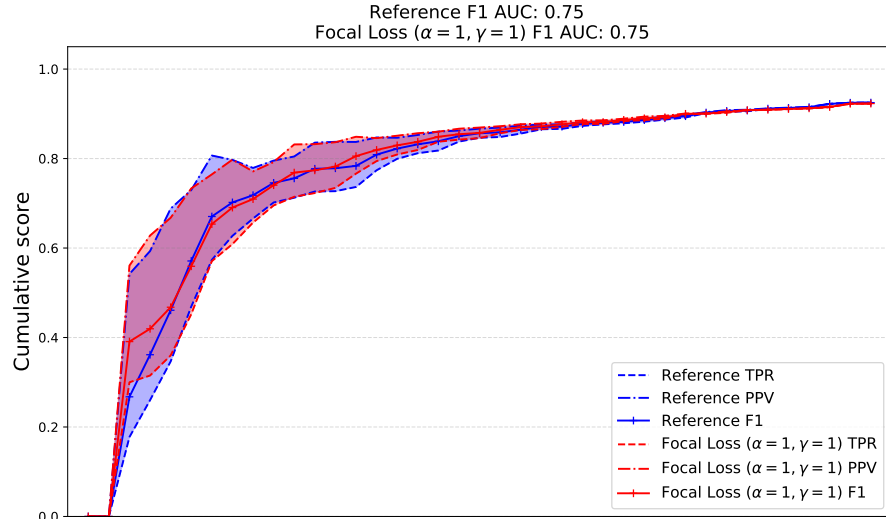


(b) Ascending F1 with  $\gamma = 2$

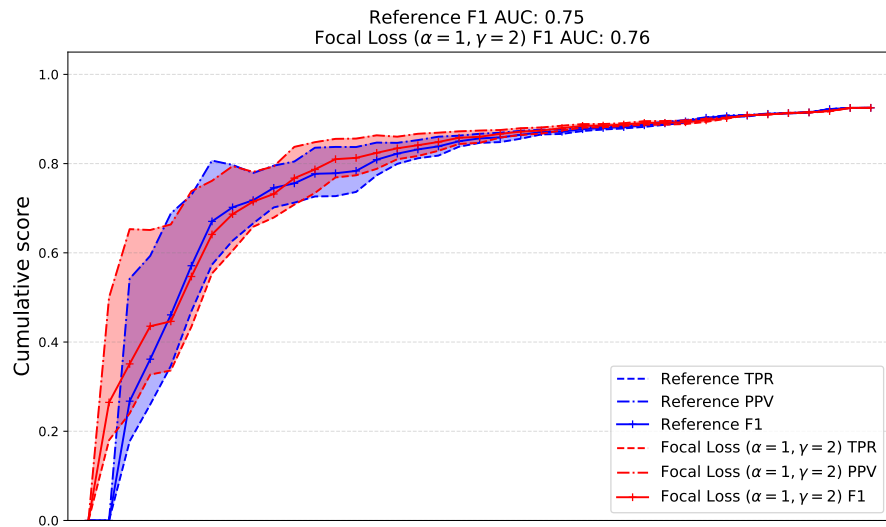


(c) Ascending F1 with  $\gamma = 4$

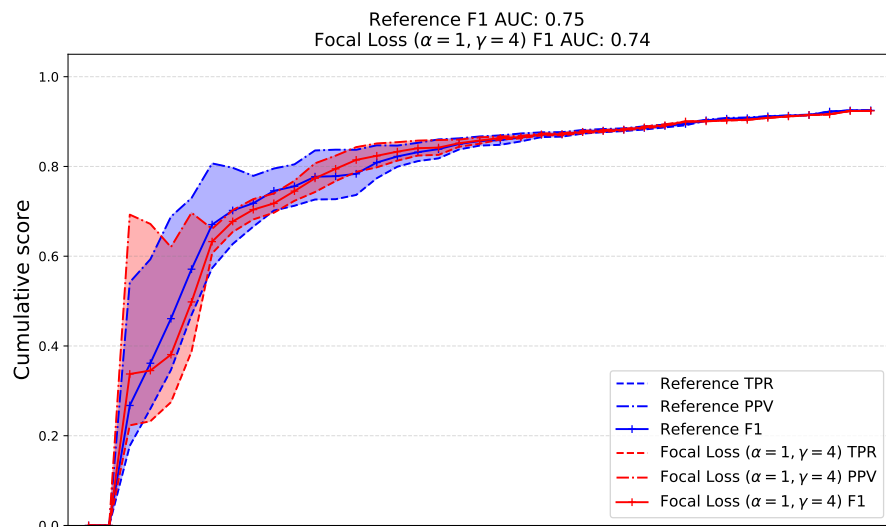
**Figure A.1.** Focal loss F1 scores plotted against reference model F1 scores in ascending order with  $\alpha = 1$ . Changing gamma has little to no difference in classifier performance



**(a)** Cumulative scores with  $\gamma = 1$

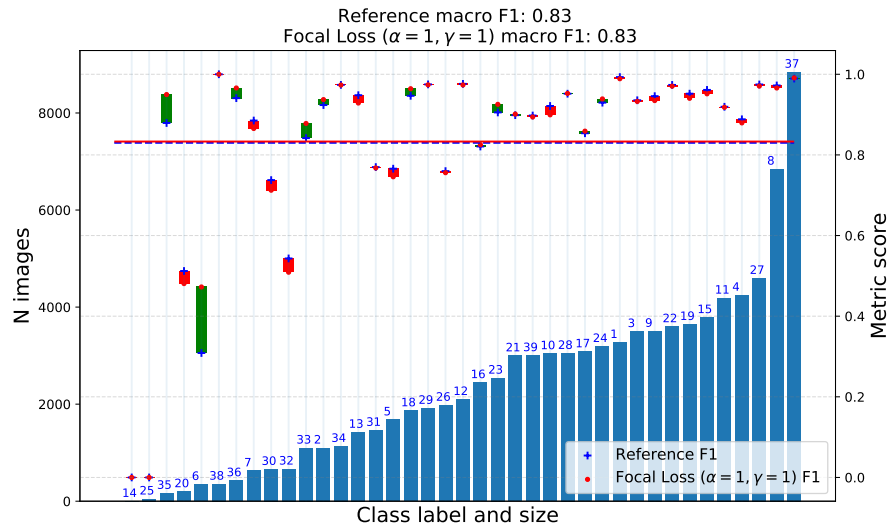
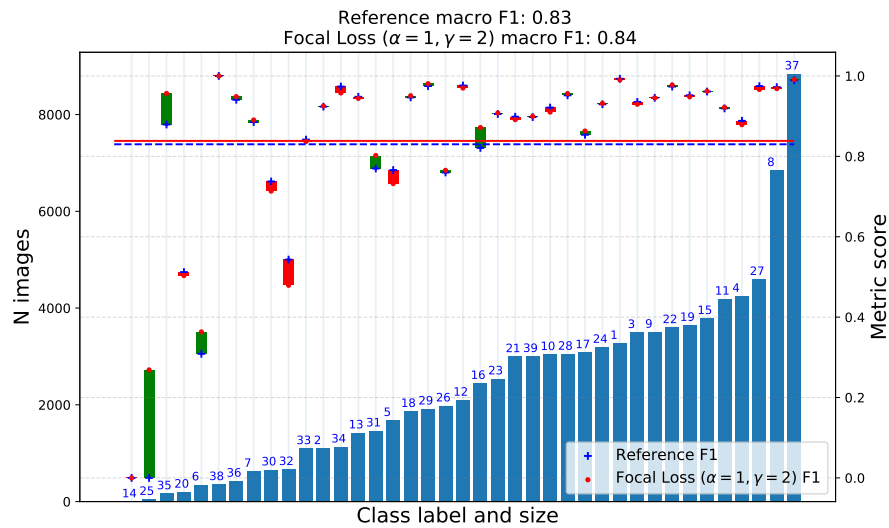
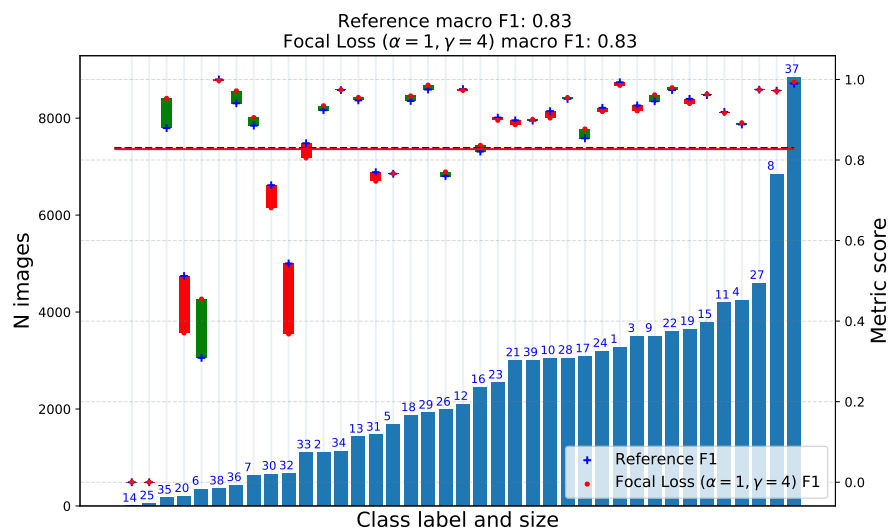


**(b)** Cumulative scores with  $\gamma = 2$



**(c)** Cumulative scores with  $\gamma = 4$

**Figure A.2.** Focal loss cumulative plots against reference model.

(a) Focal loss with  $\alpha = 1, \gamma = 1$ (b) Focal loss with  $\alpha = 1, \gamma = 2$ (c) Focal loss with  $\alpha = 1, \gamma = 4$ 

**Figure A.3.** Focal loss F1 improvements against reference. Changing gamma has vary-ing change to performance