UNIVERSITÉ DE LORRAINE | BIBLIOTHÈQUES UNIVERSITAIRES

AVERTISSEMENT

Ce document est le fruit d'un long travail approuvé par le jury de soutenance et mis à disposition de l'ensemble de la communauté universitaire élargie.

Il est soumis à la propriété intellectuelle de l'auteur. Ceci implique une obligation de citation et de référencement lors de l'utilisation de ce document.

D'autre part, toute contrefaçon, plagiat, reproduction illicite encourt une poursuite pénale.

Contact bibliothèque : ddoc-theses-contact@univ-lorraine.fr
(Cette adresse ne permet pas de contacter les auteurs)

LIENS

Code de la Propriété Intellectuelle. articles L 122. 4
Code de la Propriété Intellectuelle. articles L 335.2- L 335.10
http://www.cfcopies.com/V2/leg/leg_droi.php
http://www.culture.gouv.fr/culture/infos-pratiques/droits/protection.htm

Automatic identification of diatoms using deep learning to improve ecological diagnosis of aquatic environments

THÈSE

présentée et soutenue publiquement le 13 décembre 2023

pour l'obtention du

Doctorat de l'Université de Lorraine

(mention Ecotoxicologie, Biodiversité, Ecosystèmes)

par

Aishwarya Venkataramanan

Composition du jury

Directeur de thèse :	Dr. Philippe Usseglio-Polatera, Professor, Université de Lorraine, Metz
Co-directeurs de thèse:	Dr. Martin Laviale, Associate Professor, Université de Lorraine, Metz Dr. Cédric Pradalier, Associate Professor, GT-CNRS IRL 2958, Metz
Président :	Dr. Marianne Clausel, Professor, Université de Lorraine, Nancy
Rapporteurs :	Dr. Tim Wilhelm Nattkemper, Professor, Bielefeld University, Bielefeld Dr. Massih-Reza Amini, Professor, Université Grenoble Alpes, Grenoble
Examinateurs :	Dr. Marianne Clausel, Professor, Université de Lorraine, Nancy Dr. Soizic Morin, Research Director, INRAE, Cestas Dr. Sara Beery, Assistant Professor, MIT, Cambridge

Laboratoire Interdisciplinaire des Environnements Continentaux — UMR 7360 CNRS
International Research Lab 2958 Georgia Tech–CNRS

Mis en page avec la classe thesul.

Résumé

Les diatomées sont un type de microalgues unicellulaires trouvées dans les milieux aquatiques. Ces cellules ont une taille variant de quelques micromètres à plus de 500 micromètres de long pour les plus grandes diatomées. Elles constituent un élément significatif des communautés algales benthiques et planctoniques présentes dans le monde entier, dans divers habitats tels que les milieux d'eau douce et marins. Elles jouent un rôle crucial dans le cycle du carbone et représentent une source importante de production primaire, étant responsables de 45 % de la production primaire mondiale. De plus, elles sont également impliquées dans d'autres cycles biogéochimiques, tels que ceux de l'azote et du silicium. Par conséquent, ces organismes vitaux sont d'une importance capitale pour le fonctionnement écologique des milieux d'eau douce et marins. Ils sont également particulièrement sensibles aux changements environnementaux tels que les variations de température de l'eau, l'acidité, la disponibilité des nutriments et la pollution, ce qui en fait des indicateurs utiles pour le biomonitoring. Plusieurs normes existent pour le biomonitoring basé sur les diatomées. La motivation de cette thèse porte sur une norme spécifique utilisée pour la surveillance de la qualité des cours d'eau en France dans le cadre de la mise en œuvre de la directive-cadre sur l'eau de l'Union européenne (Conseil européen, 2000), appelée l'Indice Biologique Diatomées (IBD). L'indice est calculé sur la base des valeurs de profil écologique (c'est-à-dire la probabilité de présence dans sept classes de qualité successives) de 838 espèces clés de diatomées. Les valeurs de profil, les abondances relatives et les valeurs indicatrices des taxons de diatomées observés dans les sites d'échantillonnage sont utilisées pour calculer le score IBD afin de déterminer la qualité de l'eau.

Une des étapes essentielles pour le biomonitoring basé sur les diatomées est d'identifier la répartition des différents taxons de diatomées. Pour ce faire, les biologistes collectent périodiquement des échantillons naturels à des fins de surveillance, les traitent et identifient les espèces présentes à l'aide de microscopes. Les diatomées se composent de plusieurs taxons, identifiés en fonction des critères morphologiques tels que la forme, la taille et les ornements. Jusqu'à présent, environ 75 000 taxons de diatomées ont été identifiés, et on estime qu'il existe plus de 200 000 espèces de diatomées dans le monde.

Étant donné le grand nombre de taxons de diatomées présents, les identifier peut être une tâche ardue en termes de temps et d'efforts manuels requis. De plus, l'identification exige une expertise taxonomique, qui est difficile à trouver et est souvent sujette à des erreurs de classification en raison de facteurs tels que l'absence d'une taxonomie standardisée, les biais humains, etc. Certains de ces taxons se ressemblent beaucoup, ou les espèces au sein d'un seul taxon peuvent présenter une large diversité morphologique, ce qui rend difficile leur identification précise. Cette thèse vise à atténuer certains de ces défis en automatisant le processus d'identification à l'aide de modèles d'apprentissage en profondeur. En exploitant la puissance des algorithmes d'apprentissage en profondeur, nous développons des méthodes efficaces et précises pour identifier les taxons de diatomées. De plus, cette thèse vise à développer des méthodes pour estimer l'incertitude dans les prédictions du modèle.

L'estimation de l'incertitude est essentielle pour comprendre les limitations d'un modèle et déterminer quand des connaissances supplémentaires d'experts ou une révision manuelle peuvent être nécessaires. En développant des méthodes pour estimer l'incertitude, cette thèse vise à améliorer la fiabilité du processus d'identification.

1. Pipeline pour l'analyse automatique des diatomées

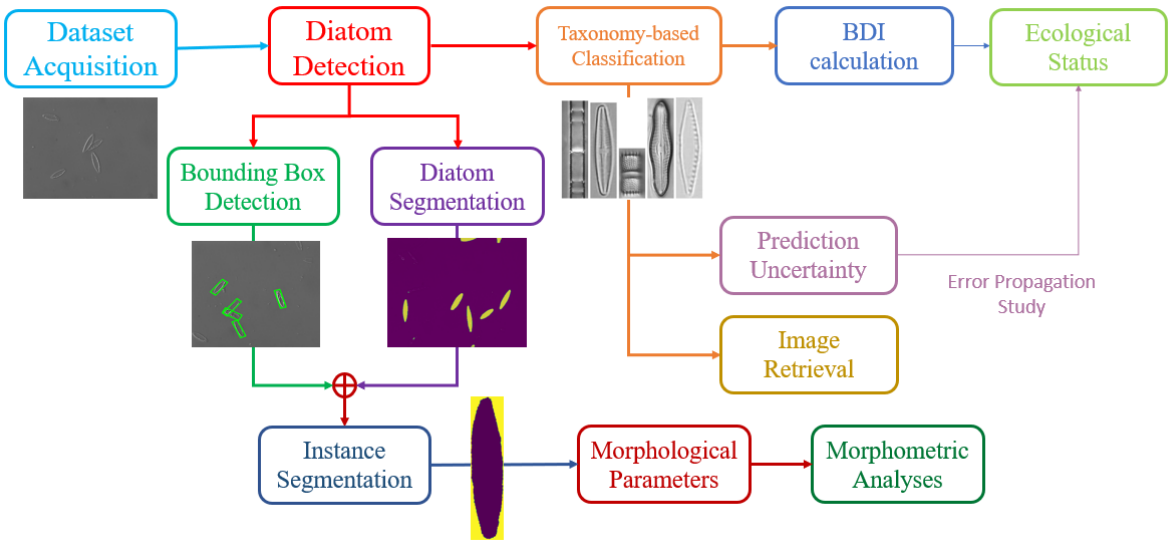


Figure 1: Cadre conceptuel illustrant les différentes étapes de la thèse.

Le pipeline global peut être résumé comme suit : Tout d'abord, les échantillons de biofilm benthique à analyser sont préparés et scannés sous le microscope pour obtenir des images microscopiques. Le réseau de détection détecte et extrait les diatomées individuelles présentes dans les images microscopiques. Les instances de diatomées détectées sont classifiées pour identifier la taxonomie. Enfin, les individus identifiés sont utilisés pour estimer le score IBD. De plus, l'incertitude dans le score IBD due à l'erreur propagée lors de chaque étape du pipeline est estimée. Le cadre conceptuel de la thèse est illustré dans la Figure 1.

1. Acquisition des données - Cette étape implique la collecte d'images microscopiques à partir des échantillons à analyser. Les échantillons sont prélevés sur les sites de surveillance et traités pour éliminer les matières organiques présentes. Ensuite, ils sont analysés sous un microscope pour acquérir les images. Deux types majeurs de microscopes sont utilisés : le microscope optique et le microscope électronique à balayage. Le travail de cette thèse est basé sur les images de microscopie optique.
2. Détection des diatomées - En général, les images de microscopie des diatomées contiennent des diatomées ainsi que des débris, qui sont de fines particules de matière organique. Parfois, les diatomées peuvent se superposer à des objets proches, ce qui entraîne l'occultation de certaines parties de leur corps.
   Il est également fréquent de voir des images où certaines parties des diatomées ne sont pas entièrement visibles ou sont cassées. Dans de telles situations, l'identité

taxonomique ne peut pas être définitivement attribuée. Il est important de filtrer ces ambiguïtés pour éviter les erreurs de prédiction. Cela est réalisé lors de l'étape de détection, où seules les diatomées entièrement visibles sont extraites.
Les algorithmes de détection se répartissent en trois catégories principales :

a. Détection par boîte englobante - Dans cette méthode, des boîtes englobantes rectangulaires sont dessinées autour de l'objet à détecter. La Figure 2 illustre ce type de détection sur des images de microscopie des diatomées, avec des boîtes rectangulaires vertes dessinées autour des diatomées détectées. La détection par boîte englobante peut être effectuée soit sous forme alignée sur les axes, soit sous forme de boîtes englobantes rotatives. L'image de gauche dans la Figure 2 est un exemple de détection alignée sur les axes, et l'image de droite est une détection par boîte englobante rotative.

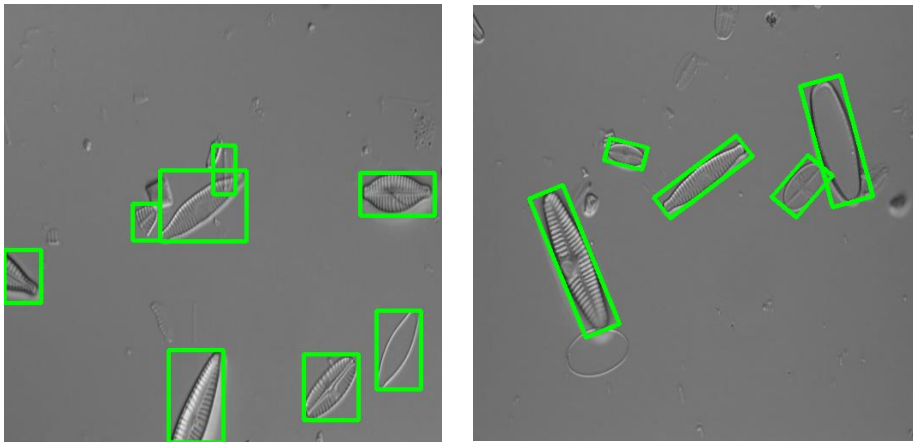


Figure 2: Exemples de détection de boîtes englobantes pour les diatomées. (À gauche) Détection de boîtes englobantes horizontales, (À droite) Détection de boîtes englobantes rotatives.

b. Segmentation sémantique - Dans la segmentation sémantique, chaque pixel de l'image est attribué à l'une des étiquettes de classe prédéfinies. La Figure 3 fournit deux exemples de segmentation sémantique des diatomées. Étant donné une image de microscopie en entrée, comme celles montrées à gauche, les régions jaunes sur les images de droite représentent les diatomées et les régions violettes représentent les zones non diatomiques. Ce type de détection met en évidence la forme des objets détectés.

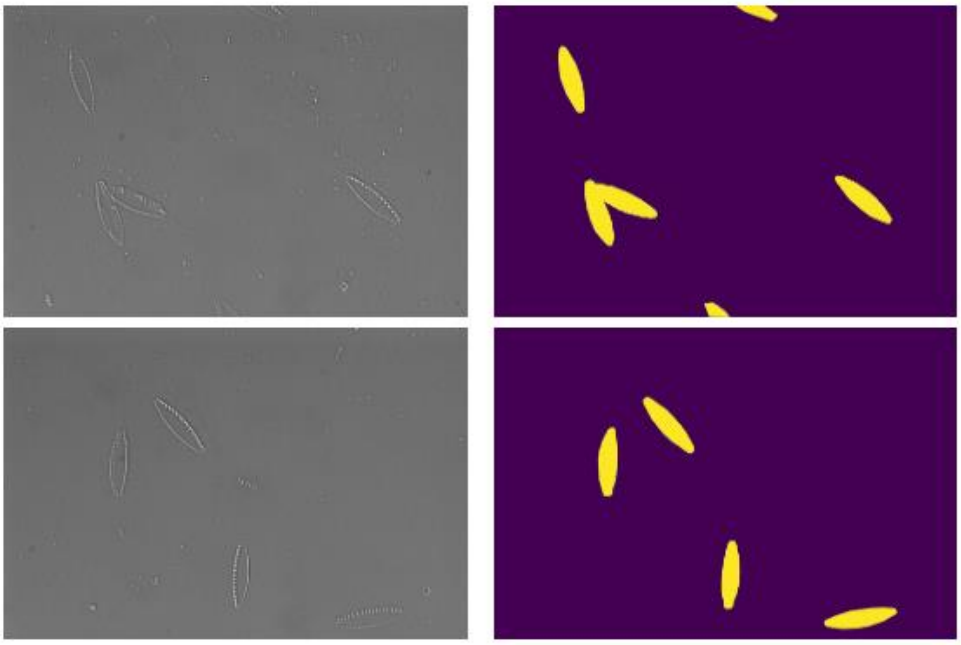


Figure 3: Exemple de segmentation sémantique des diatomées. (À gauche) Images de microscopie des diatomées (À droite) Étiquettes de segmentation sémantique. Les régions jaunes représentent les diatomées, tandis que les régions violettes représentent les zones autres que les diatomées.

c. Segmentation d'instance - Elle implique l'identification d'objets individuels dans une image et leur segmentation, c'est-à-dire l'attribution de chaque pixel de l'image à une instance d'objet spécifique. Contrairement à la segmentation sémantique, qui attribue chaque pixel à une étiquette de classe, la segmentation d'instance différencie également les objets appartenant à la même classe en attribuant un identifiant unique à chaque instance.

La segmentation d'instance s'avère particulièrement précieuse dans la détection des diatomées lorsque plusieurs diatomées sont présentes dans une image. Elle permet la détection et la segmentation précises de chaque instance individuelle de diatomée, qui peuvent ensuite être utilisées pour des tâches ultérieures telles que le comptage des diatomées et l'extraction des paramètres morphologiques des diatomées non superposées. Cela est ensuite utilisé pour des analyses morphométriques.

3. Classification basée sur la taxonomie - Chacune des diatomées détectées est identifiée en fonction de sa taxonomie. Un défi majeur dans la classification des diatomées est que les individus appartenant à des taxons différents peuvent avoir une apparence visuelle très similaire, c'est-à-dire une grande similarité interspécifique. En même temps, les individus appartenant au même taxon peuvent présenter des changements d'apparence drastiquement différents, c'est-à-dire une grande variance intraspécifique. Cela rend la tâche de classification difficile et sujette à des erreurs de prédiction. Une contribution de cette thèse est de répondre à ce défi et de développer des méthodes pour réduire les erreurs dans les prédictions taxonomiques. De plus, nous développons une méthode pour estimer l'incertitude dans la prédiction du réseau de classification, ce qui peut fournir des informations précieuses sur la fiabilité de la sortie du modèle.

4. Calcul du BDI - La taxonomie des diatomées identifiées, leur abondance (le nombre d'individus dans chaque taxon) et leurs profils écologiques récupérés à partir de bases de données nationales sont utilisés pour estimer la valeur de l'IBD, ce qui permet de déterminer l'état écologique des masses d'eau. Dans ce travail, nous étudions l'incertitude due à la propagation des erreurs dans le pipeline sur la valeur de l'IBD et analysons sa robustesse.
5. Récupération d'images - Étant donné une image de requête, le système de récupération d'images extrait des images similaires de la base de données des diatomées. Ce processus aide les utilisateurs à trouver rapidement des images visuellement ou contextuellement liées à l'image de requête, réduisant ainsi les correspondances potentielles et fournissant un point de départ pour des investigations ultérieures.

2. Les ensembles de données

Le travail actuel utilise des modèles d'apprentissage en profondeur pour la détection et la classification taxonomique. Une condition préalable cruciale pour entraîner ces modèles est la disponibilité de vastes ensembles de données étiquetées comprenant des milliers d'échantillons. Ces ensembles de données d'entraînement se composent de données d'entrée et de leurs prédictions de vérité terrain correspondantes. Le modèle d'apprentissage en profondeur apprend à partir de ces ensembles de données pour faire des prédictions lorsqu'il est confronté à de nouvelles données en entrée.

Lors de l'entraînement des modèles d'apprentissage en profondeur, il est courant de disposer de trois ensembles de données étiquetées : entraînement, validation et test. Le modèle est entraîné à l'aide de l'ensemble d'entraînement. L'ensemble de validation est utilisé pour ajuster les hyperparamètres du modèle, tandis que l'ensemble de test est utilisé pour effectuer des prédictions et évaluer les performances du modèle. Puisque le modèle apprend à partir des données d'entraînement, celles-ci devraient couvrir tous les types d'entrées possibles que le réseau rencontrera lors des tests.

Le processus de sélection des ensembles de données pour cette thèse revêtait une importance significative, car il était essentiel de garantir leur alignement avec les objectifs spécifiques du projet et son contexte global. Le principal critère de sélection était la technique de microscopie prédominante utilisée dans les pratiques de biomonitoring réglementaires en France, à savoir la microscopie optique en contraste de phase différentiel (DIC - Differential Interference Contrast). Il est à noter qu'aucun ensemble de données d'images de diatomées DIC n'était disponible publiquement pour l'entraînement des modèles d'apprentissage automatique. Pour combler cette lacune, nous avons développé des méthodes pour extraire des images de diatomées à partir d'atlas disponibles publiquement, développés par des praticiens, et pour générer des ensembles de données synthétiques, mettant en valeur l'originalité de cette contribution.

Cependant, reconnaissant la communauté plus vaste d'experts en diatomées utilisant la technique de champ clair (brightfield), nous avons établi une collaboration avec des chercheurs de l'Université de Duisburg-Essen (https://www.uni-due.de/phycology/). Cette collaboration

visait à élargir la portée de nos modèles en les testant sur leur ensemble de données, qui devrait bientôt être rendu disponible. De plus, afin de démontrer la polyvalence des méthodes développées pour diverses applications, nous avons effectué des évaluations sur des ensembles de données standard en vision par ordinateur.

2.1 Ensembles de données de diatomées

Images microscopiques DIC complètes

Pour la détection, les images de microscopie ont été obtenues à partir de deux types d'échantillons de diatomées :

1. CLEURIE - Ce sont des échantillons naturels collectés dans le ruisseau de Cleurie, dans les Vosges, en France, sur des substrats minéraux durs et fixés en ajoutant de l'éthanol, montrant des communautés naturelles composées de diverses espèces de diatomées. Cet ensemble de données présente une variété diversifiée de taxons de diatomées, avec une morphologie largement variable.

2. NPAL - Ce sont des échantillons prélevés à partir d'une culture en laboratoire de la diatomée Nitzchia palea (NPAL). Les images contiennent un seul taxon de diatomée, et donc la morphologie n'est pas aussi variée que celle de l'ensemble de données CLEURIE, mais ces individus pourraient présenter différentes variations morphologiques intra-classe.

Chaque échantillon liquide a été soumis à une oxydation à l'aide de peroxyde d'hydrogène chaud afin de retirer la matière organique, rincé à l'eau déionisée, puis monté sur des lames de microscope permanentes avec du Naphraxinto, une résine à indice de réfraction élevé pour être visualisé à un grossissement de 1000x à l'aide d'un microscope optique.

Images individuelles DIC - Ensemble de données Atlas

Le réseau de classification prend des instances individuelles de diatomées et prédit le taxon correspondant. L'ensemble de données Atlas se compose d'images de diatomées extraites des atlas de diatomées en libre accès appartenant à cinq régions différentes : Rhône-Alpes, Île-de-France, Bourgogne, Loire et Provence-Alpes-Côte d'Azur. À partir des atlas au format PDF, les images de diatomées ont été extraites automatiquement à l'aide du scraping PDF. Il s'agit d'un moyen économique et efficace d'obtenir un ensemble de données de classification et de détection basé sur la taxonomie, puisque les images individuelles de diatomées sont librement accessibles, accompagnées de leur taxonomie pour les étiquettes de vérité terrain. À des fins d'entraînement, nous nous sommes concentrés sur les taxons disposant d'un minimum de 30 images de diatomées disponibles. Par conséquent, l'ensemble de données était composé de 197 taxons de diatomées, comprenant une importante collection de 14 694 images. Ces images ont été redimensionnées à une taille uniforme de 256x256 pixels. De plus, pour garantir l'exactitude de la taxonomie au sein de l'ensemble de données, des diatomistes experts ont minutieusement vérifié et validé les classifications taxonomiques. Le nombre maximum d'images par taxon étant de 504 et le nombre minimum d'images par taxon étant de 30.

Debris

Le jeu de données "Debris" se compose de 600 images représentant une variété diversifiée d'objets de débris, extraites à partir d'images microscopiques.

Les images en champ clair - Le jeu de données des Diatomées de l'UDE

Ce jeu de données a été obtenu en collaboration avec le Dr Bank Beszteri et le Dr Michael Kloster de l'Université de Duisburg-Essen, en Allemagne. Il comporte 320 échantillons benthiques provenant de 14 écosystèmes fluviaux/lacustres différents et d'une expérience multi-stress. Les spécimens sur lame ont été numérisés en format numérique à l'aide d'un microscope à balayage de lames VS200 en mode de champ clair, en utilisant un objectif d'immersion à l'huile UPLXAPO60XO 60x/1.42. Le jeu de données se compose de 83 354 images de diatomées en microscopie optique. Nous considérons deux sous-ensembles du jeu de données pour les expériences : (i) D25 : ce jeu de données comprend les taxons avec un nombre d'échantillons ≥ 25, comprenant 238 taxons (ii) D50 : ce jeu de données comprend les taxons avec un nombre d'échantillons ≥ 50, comprenant 176 taxons.

Autres ensembles de données

Les ensembles de données suivants ont été utilisés pour évaluer les modèles développés : WHOI-Plankton, CIFAR10, CIFAR100, FashionMNIST, SVHN et CUB-200-2011.

Résumé

Les diatomées sont un type d'algues unicellulaires que l'on trouve dans tous les environnements aquatiques. Ces organismes sont très sensibles aux changements de qualité de l'eau et des conditions environnementales. Cette caractéristique les rend utiles pour la bioindication : en France, l'Indice Biologique Diatomées (IBD) est utilisé de manière courante depuis l'an 2000 pour évaluer la qualité écologique des cours d'eau, dans le cadre de la Directive-Cadre sur l'Eau de l'Union Européenne. Traditionnellement, l'évaluation de la qualité écologique de l'eau et de la santé des écosystèmes à travers les diatomées est un processus méticuleux et chronophage. À partir d'échantillons naturels, des experts qualifiés en taxonomie des diatomées identifient les individus jusqu'au niveau de l'espèce grâce à des observations au microscope optique. Cependant, ce processus d'identification manuel n'est pas sans ses défis. Des facteurs tels que la qualité de l'équipement de microscopie, le niveau d'expertise des opérateurs, et la subjectivité inhérente au jugement humain contribuent tous à la variabilité des résultats d'identification. Dans ce contexte, ce travail vise à réduire cette variabilité en automatisant le processus d'identification des diatomées. Le premier objectif est le développement d'un outil pour détecter les diatomées dans les images de microscope, en les distinguant parmi une multitude d'autres objets présents. Le deuxième objectif est de classifier les diatomées détectées jusqu'au niveau de l'espèce, avec un certain niveau de confiance. La principale contribution de cette recherche réside dans la création d'une chaîne de traitement de bout en bout pour automatiser l'identification des diatomées dans les images de microscopie DIC (Microscope à contraste interférentiel), basée sur des méthodes d'apprentissage profond. L'apprentissage profond a démontré des capacités remarquables dans les tâches nécessitant une reconnaissance de motifs complexes et une classification, ce qui le rend idéalement adapté à l'automatisation du processus nuancé et complexe d'identification des diatomées. Cependant, l'adoption de l'apprentissage profond n'est pas sans ses propres défis. Une exigence fondamentale était l'accès à des ensembles de données substantiels et correctement annotés pour former efficacement ces réseaux neuronaux. Pour surmonter ce défi, une méthode de génération de jeux de données synthétiques a été proposée en utilisant des atlas de diatomées disponibles publiquement. En créant méticuleusement des

données artificielles tout en préservant les caractéristiques des images réelles de diatomées, cette méthode a complété les données d'entraînement disponibles et amélioré la capacité du réseau neuronal à généraliser son apprentissage à de nouveaux échantillons de diatomées. Un autre défi est apparu en raison des similarités entre différentes espèces de diatomées (similarité interspécifique) et des variations au sein d'une même espèce (variabilité intraspécifique) pendant le processus de classification automatisée. Cela constituait un défi considérable pour la classification précise des diatomées. Ce travail a exploré des stratégies et des techniques innovantes pour tenir compte de ces complexités, améliorant ainsi la capacité du système à discerner des différences subtiles entre espèces et à effectuer des classifications précises. Enfin, une nouvelle méthode de quantification de l'incertitude dans les classificateurs profonds a également été développée, ce qui a contribué à améliorer la fiabilité du processus de classification, en particulier en permettant de détecter les images hors distribution. Pour démontrer la praticabilité de l'outil développé, son application en biomonitoring a été présentée en calculant l'IBD sur un grand nombre d'échantillons représentatifs du bassin du Rhin-Meuse. En tirant parti de la puissance des réseaux neuronaux profonds, cette chaîne de traitement rationalise et accélère le processus d'identification, offrant une alternative plus efficace aux méthodes de classification manuelles.

Abstract

Diatoms are a type of unicellular algae found in all aquatic environments. These organisms are very sensitive to changes in water quality and habitat conditions. This characteristic makes them useful for bioindication: In France, the Biological Diatom Index (BDI) has been used routinely since 2000 to assess the ecological quality of watercourses, within the framework of the European Water Framework Directive. Traditionally, assessing the ecological quality of water and ecosystem health through diatoms is a meticulous and time-consuming process. From natural samples, qualified experts in diatom taxonomy identify individuals to the species level based on optical microscope observations. However, this manual identification process is not without its challenges. Factors such as the quality of microscopy equipment, the level of expertise of users, and the inherent subjectivity of human judgment all contribute to variability in the identification results. In this context, this work aims to reduce this variability by automating the diatom identification process. The first objective is the development of a tool to detect diatoms within microscope images, distinguishing them among a myriad of other objects present. The second objective is to classify the detected diatoms down to the species level, with a certain level of confidence. The main contribution of this research lies in the creation of an end-to-end pipeline for automating diatom identification in DIC (Differential interference contrast) microscopy images, based on deep learning methods. Deep learning has demonstrated remarkable capabilities in tasks requiring complex pattern recognition and classification, which is ideally suited for automating the nuanced and complex process of diatom identification. However, the adoption of deep learning is not without its own challenges. A fundamental requirement was access to substantial and properly annotated datasets to effectively train these neural networks. To overcome this challenge, a method for generating synthetic datasets was proposed using publicly available diatom atlases. By meticulously crafting artificial data while preserving the characteristics of real diatom images, this method supplemented available training data and improved the neural network's ability to generalize its learning to new diatom samples. Another challenge arose from the similarities between different diatom species (inter-species similarity) and variations within a single species (intra-species variability) during the automated classifica-

tion process. This posed a considerable challenge to the accurate classification of diatoms. This work explored innovative strategies and techniques that account for these complexities, thereby improving the system's ability to discern subtle differences and make accurate classifications. Finally, a new method for quantifying uncertainty in deep classifiers was also developed, which contributed to improving the reliability of the classification process, in particular by making it possible to detect out-of-the-distribution images. To demonstrate the practicability of the developed tool, its application in biomonitoring was presented by calculating the BDI on a large number of representative samples from the Rhine-Meuse basin. By harnessing the power of deep neural networks, this pipeline streamlines and accelerates the identification process, providing a more efficient alternative to manual classification methods.

To my parents.

Acknowledgements

Embarking on this journey towards obtaining a PhD has been a transformative experience, and it wouldn't have been possible without the invaluable support and guidance of numerous individuals. First and foremost, I am deeply grateful to my advisors, Dr. Philippe Usseglio-Polatera, Dr. Martin Laviale, and Dr. Cédric Pradalier, for not only believing in me but also for their unwavering support, encouragement, and scholarly insights throughout the entirety of my doctoral pursuit. Their mentorship has been instrumental in shaping my academic journey and research endeavors.

I owe an immense debt of gratitude to my parents for their unending love, encouragement, and sacrifices. Their unwavering belief in my potential has been my anchor during the challenging times of this academic pursuit. I am forever indebted to them for their boundless support and motivation.

I extend my heartfelt appreciation to my colleagues at the Dream Lab, whose camaraderie and intellectual exchanges have enriched my research experience. In particular, I am thankful to Assia Benbihi and Antoine Richard for their guidance during both my Master's and PhD theses. Their insights and mentorship have been invaluable pillars in shaping my research path.

Furthermore, I extend my thanks to my colleagues from Georgia Tech Europe and LIEC, along with co-authors, interns, and master thesis students, whose collaborative efforts and contributions have significantly enriched the scope and depth of my research project. Their diverse perspectives and collaborative spirit have truly been enlightening and enriching.

Lastly, I express my sincere gratitude to my family and friends, whose support, understanding, and encouragement have been a source of strength and motivation throughout this journey. Their belief in me has been an invaluable source of inspiration.

Each of these individuals has played an integral role in shaping my academic and personal growth, and for that, I am immensely grateful.

Contents

1 Introduction **1**

1.1 Pipeline for automatic diatom analysis . . . 3

1.2 Datasets . . . 6

1.2.1 Diatom Datasets . . . 7

1.2.2 Other Datasets . . . 9

1.2.3 Dataset annotation . . . 10

1.3 Overview of the thesis . . . 10

1.4 Contributions . . . 12

2 Concepts and Related Works **15**

2.1 A Brief Introduction to Learning Algorithms . . . 15

2.1.1 Some jargons in machine learning . . . 17

2.2 Deep Learning Concepts . . . 17

2.2.1 Model Components . . . 18

2.3 Classification . . . 21

2.4 Bounding Box Detection . . . 24

2.4.1 One Stage Networks . . . 25

2.4.2 Two Stage Networks . . . 26

2.4.3 Rotated Bounding box detection . . . 28

2.5 Semantic Segmentation . . . 28

2.6 Hierarchies for Identification . . . 30

3 Diatom Detection **34**

3.1 Introduction . . . 34

3.2 Related Works . . . 36

3.2.1 Detection methods for aquatic organisms . . . 36

3.2.2 Training Deep CNNs with scarse data . . . 38

3.3 Synthetic Dataset Generation . . . 39

3.4 Experiments on bounding box detection . . . 41
3.4.1 Sensitivity Analysis of Parameters in Synthetic Dataset Generation . 41
3.4.2 Results . . . 44
3.5 Experiments on automatic diatom segmentation . . . 51
3.5.1 Materials and Methods . . . 52
3.5.2 Experiments . . . 54
3.5.3 Results . . . 55
3.6 Conclusion . . . 59

4 Classification and Uncertainty Estimation 60
4.1 Tackling Inter-class Similarity and Intra-class Variance for Diatom Classification 60
4.1.1 Related Works . . . 62
4.1.2 Method . . . 64
4.1.3 Experiments . . . 66
4.1.4 Results . . . 68
4.1.5 Discussion . . . 70
4.1.6 Conclusion . . . 71
4.2 Uncertainty Estimation in Classification . . . 71
4.2.1 Related Works . . . 73
4.2.2 Method . . . 75
4.2.3 Experiments on Diatom Dataset . . . 79
4.2.4 Experiments on Standard Datasets . . . 82
4.2.5 Discussion . . . 93
4.2.6 Conclusion . . . 93
4.3 Conclusion on Diatom Classification . . . 94

5 Misclassification Analysis in BDI 95
5.1 Introduction . . . 95
5.2 Material and Methods . . . 97
5.2.1 Generation of the synthetic diatom inventory dataset . . . 97
5.2.2 BDI and EQR calculations . . . 99
5.2.3 Deep learning classifier network . . . 100
5.2.4 Robustness analysis of the deep learning classifier . . . 101
5.2.5 Misclassification analysis . . . 102
5.3 Results . . . 103
5.3.1 Representativeness of the subset of species available for simulating new inventories . . . 103

5.3.2 Classifier Performance . . . 105
5.3.3 Robustness analysis . . . 107
5.3.4 Misclassification analysis . . . 107
5.4 Discussion and Conclusion . . . 110

6 Hierarchy for Image Retrieval **112**
6.1 Introduction . . . 112
6.2 Related Works . . . 114
6.3 Method . . . 115
6.3.1 Hierarchy Construction . . . 115
6.3.2 Distance Calculation . . . 116
6.3.3 Robustness Analysis . . . 117
6.4 Experiments . . . 117
6.4.1 Baselines . . . 117
6.4.2 Experimental Setup . . . 118
6.4.3 Evaluation Metrics . . . 119
6.5 Results . . . 119
6.6 Conclusion . . . 124

7 Conclusion and Future Work **125**
7.1 Acknowledgement . . . 128

Bibliography **129**

A Atlas Dataset **150**

B Morphological Parameters Formulas **160**

C Supplementary Materials for Chapter 5 **162**

Chapter 1

Introduction

Diatoms are a type of unicellular microalgae found in aquatic environments. These cells have size ranging from a few micrometers to over 500 micrometers in length for the largest diatoms. They are a significant component of both benthic and planktonic algal communities found across the globe in various habitats, including freshwater, and marine environments. They play a crucial role in the carbon cycle and are an important source of primary production, responsible for 45% of the global primary production. Moreover, they are also involved in other biogeochemical cycles, such as nitrogen and silicon. Therefore, these vital organisms are of utmost significance for the ecological functioning of freshwater and marine environments [11, 161]. They are also particularly susceptible to environmental changes such as alterations in water temperature, acidity, nutrient availability, and pollution, making them useful for biomonitoring [35]. Several standards exist for diatom-based biomonitoring [39,87]. The motivation of this thesis is on a specific standard used for surveillance of water course quality in France under the implementation of the European Water Framework Directive (European Council, 2000), called the Biological Diatom Index (BDI) [35]. The index is calculated based on the ecological profile values (i.e. the probability of presence in seven successive quality classes) of 838 key diatom species. The profile values, relative abundances and the indicator values of the diatom taxa observed in sampling sites, are used to calculate the BDI score to determine the quality of water.

One of the essential steps for diatoms-based biomonitoring is to identify the distribution of occurrence of the different diatom taxa. To do this, biologists periodically collect natural samples for monitoring, process them, and identify the species present using microscopes. Diatoms consist of several taxa, that are identified based on the morphological criteria exhibited, such as the shape, size, and ornamentations. So far, approximately 75,000 diatom taxa have been identified, and it is estimated that there are over 200,000 diatom species in the world [11, 122]. Figure. 1.1 provides some examples of diatom taxa exhibiting varying

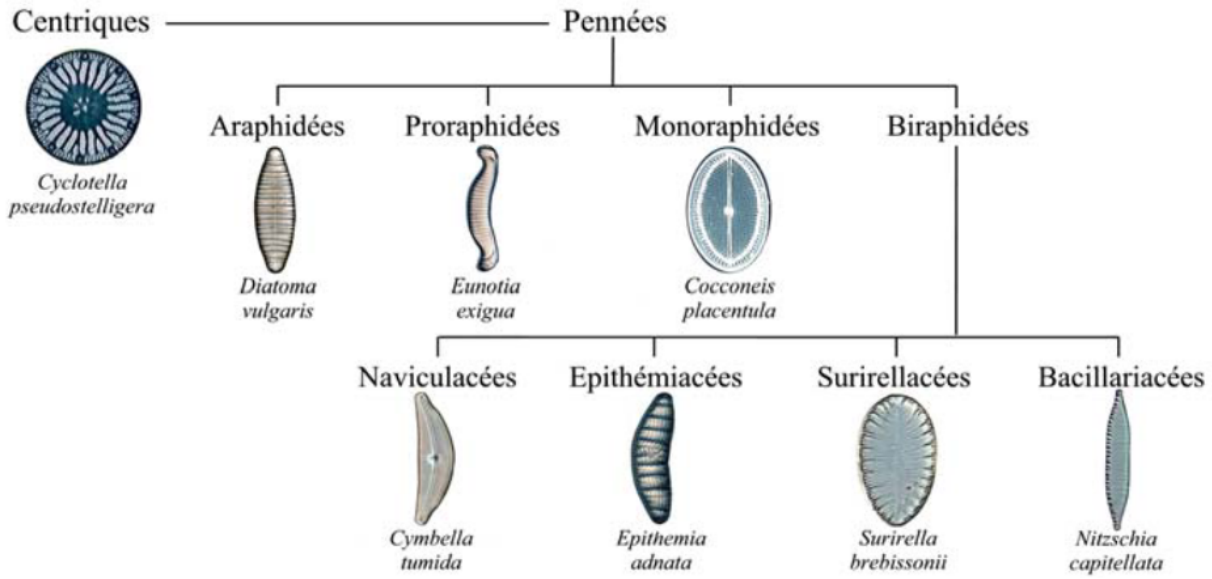


Figure 1.1: Benthic diatoms categorized into a simple taxonomic hierarchy based on their exhibited morphological features. Source: [128]

morphological features.

Given the large number of diatom taxa present, identifying them can be a strenuous task in terms of the time and manual effort required. Moreover, identification requires taxonomic expertise, which is difficult to find and is often subject to misclassifications due to factors such as the lack of a standardized taxonomy [105], human biases, etc. Some of these taxa look very similar to each other, or the species within a single taxon can exhibit wide morphological diversity, making it challenging to identify all of them accurately. This thesis aims to alleviate some challenges by automating the identification process using deep learning models. By leveraging the power of deep learning algorithms, we develop efficient and accurate methods for identifying diatom taxa. Furthermore, this thesis aims to develop methods to estimate the uncertainty in model predictions. Uncertainty estimation is critical for understanding the limitations of a model and determining when additional expert knowledge or manual review may be necessary. By developing methods for estimating uncertainty, this thesis aims to improve the reliability of the identification process.

The rest of this chapter is organized as follows: Section 1.1 illustrates the overall pipeline. Section 1.2 details the datasets that were used in developing and evaluating the models, Section 1.3 provides the outline of the thesis structure, and Section 1.4 lists the contributions of this thesis.

1.1 Pipeline for automatic diatom analysis

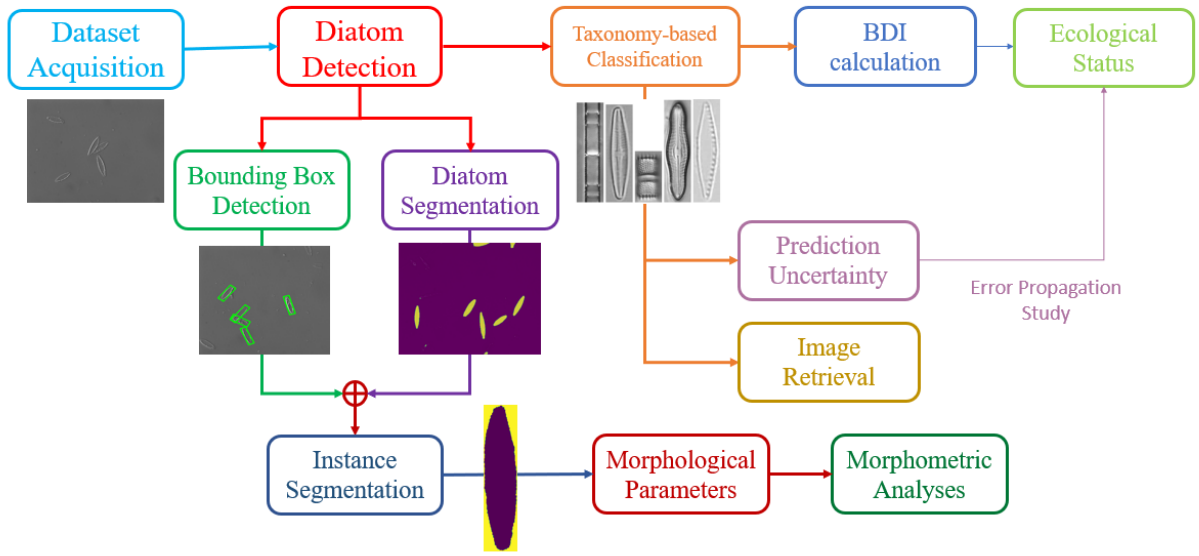


Figure 1.2: Conceptual framework illustrating the different steps of the thesis.

This section provides a high level overview of the thesis. The overall pipeline can be summarized as follows: First, benthic biofilm samples to be analysed are prepared and scanned under the microscope to obtain microscopy images. The detection network detects and extracts the individual diatoms present in the microscopy images. The detected diatom instances are classified to identify the taxonomy. Finally, the identified individuals are used to estimate the BDI score. In addition, the uncertainty in BDI due to error propagated during each step of the pipeline is estimated. The conceptual framework of the thesis is illustrated in Figure. 1.2.

1. **Dataset Acquisition -** This step involves collecting microscopy images from the samples to be analysed. Samples are collected from the monitoring sites and processed to remove organic materials present. Then, they are analysed under a microscope to acquire the images. There are two major types of microscopes used: light microscope and scanning electron microscope. The work in this thesis is based on the light microscopy images.

2. **Diatom Detection -** Typically, the diatom microscopy images contain diatoms and debris, which are fine particulated organic matter particles. Sometimes, the diatoms can overlap with nearby objects, resulting in parts of its body being occluded. It is

also common to see images where parts of the diatoms are not fully visible or broken. In situations like these, the taxonomic identity cannot be definitively assigned. It is important to filter out these ambiguities to avoid mispredictions. This is performed in the detection stage, where, only the fully visible diatoms are extracted.

Detection algorithms fall into three main categories:

(a) Bounding box detection – In this method, rectangular bounding boxes are drawn around the object to be detected. Figure. 1.3 illustrates this type of detection on diatom microscopy images, with green rectangular boxes drawn around the detected diatoms. Bounding box detection can be performed either as an axis-aligned or a rotated type. The left image in Figure. 1.3 is an example of the axis-aligned detection, and the right image is a rotated bounding box detection.

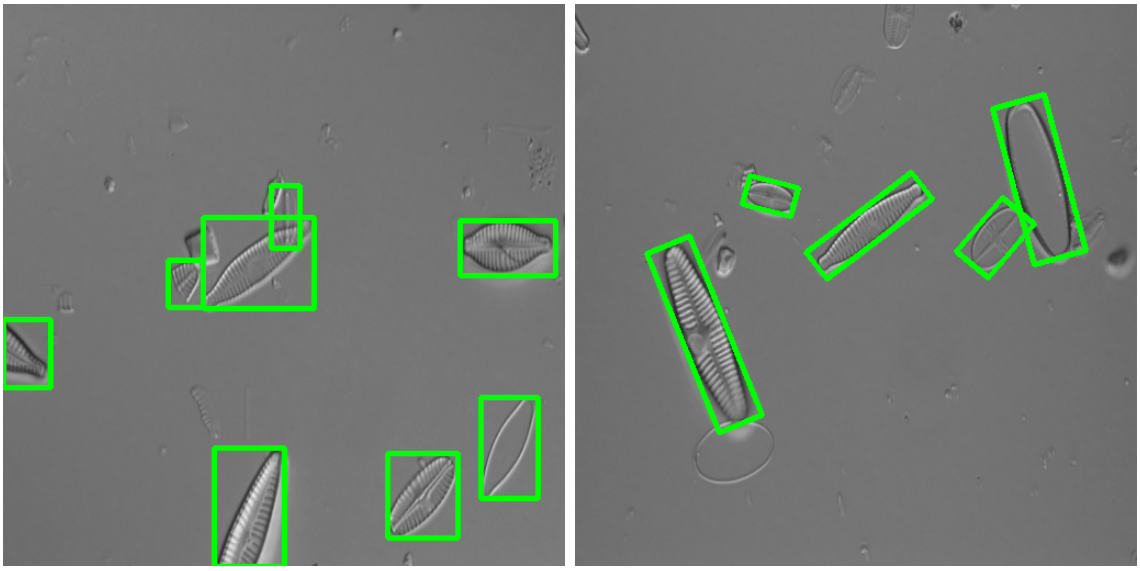


Figure 1.3: Examples of bounding box detection for diatoms. (Left) Horizontal bounding box detection, (Right) Rotated bounding box detection.

(b) Semantic Segmentation — In semantic segmentation, every pixel in the image is assigned one of the pre-defined class labels. Figure. 1.4 provides two examples of semantic segmentation of diatoms. Given an input microscopy image, such as the ones shown on the left, the yellow regions on the right hand side images represent the diatoms and the violet regions represent the non-diatom areas. This type of detection brings out the shape of the detected objects.

(c) Instance Segmentation – It involves identifying individual objects within an image and segmenting them, i.e., assigning each pixel of the image to a specific object instance. In contrast to semantic segmentation, which assigns each pixel to a class

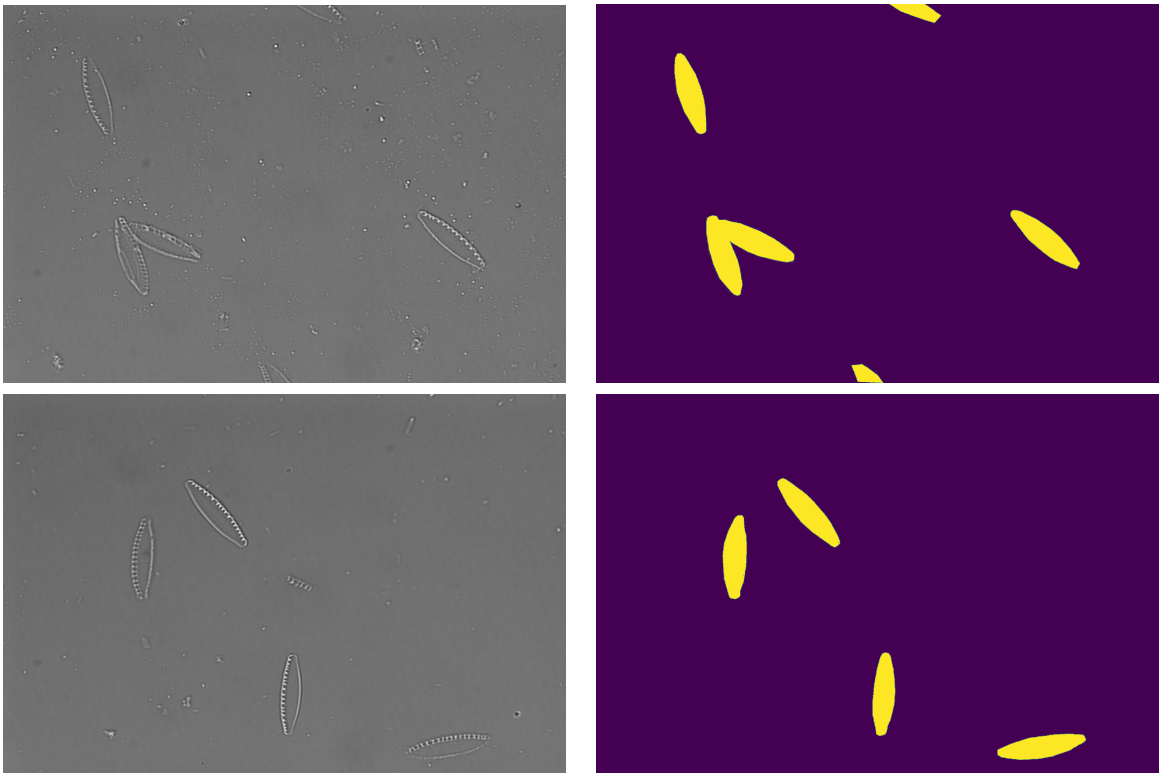


Figure 1.4: Example of semantic segmentation of diatoms. (Left) Microscopy images of diatoms (Right) Semantic segmentation labels. The yellow regions represent the diatoms, and the violet regions represent regions other than diatoms.

label, instance segmentation also differentiates between objects belonging to the same class by assigning a unique identifier to each instance.

Instance segmentation proves especially valuable in diatom detection when multiple diatoms are present within an image. It allows for the precise detection and segmentation of each individual diatom instance, which can then be leveraged for subsequent tasks such as diatom counting and extracting **morphological parameters** of the non-overlapping diatoms. This is further used for **morphometric analyses**.

3. **Taxonomy-based Classification -** Each of the detected diatoms are identified based on their taxonomy. One major challenge in diatom classification is that individuals belonging to different taxa can have very similar visual appearance *i.e.,* high inter-species similarity. At the same time, individuals belonging to the same taxon can display drastically different appearance changes *i.e.,* has high intra-species variance. This makes the classification task challenging, subject to mispredictions. One contribution of this thesis is to address this and develop methods to reduce the errors in taxonomic predictions. Additionally, we develop a method to estimate the **uncertainty** in the classification

network's prediction, which can provide valuable information about the reliability of the model's output.

4. **BDI calculation -** The taxonomy of the identified diatoms, their abundance (the number of individuals in each taxon) and their ecological profiles retrieved from national databases are used to estimate the BDI value, which allows determining the **ecological status** of water bodies. In this work, we study the uncertainty due to error propagation in the pipeline on the BDI value and analyse its robustness.

5. **Image Retrieval -** Given a query images, the image retrieval system retrieves similar images from the diatom database. This process helps users quickly find images that are visually or contextually related to the query image, narrowing down the potential matches and providing a starting point for further investigation.

1.2 Datasets

The present work employs deep learning models for detection and taxonomic classification. A crucial prerequisite for training these models is the availability of large, labelled datasets comprising thousands of samples. These training datasets consist of input data and their corresponding ground truth predictions. The deep learning model learns from these datasets to make predictions when provided with a new input.

While training deep learning models, it is customary to have three sets of labelled data: *train*, *validation* and *test*. The model is trained using the train set. The validation set is used to tune the hyperparameters of the model, and the test set is used for making predictions and for evaluating the model's performance. Since the model learns from the training data, it should cover all possible input types that the network will encounter during testing.

The process of selecting datasets for this thesis held significant importance, as it was imperative to ensure alignment with the project's specific objectives and overall context. The primary determinant in this selection was the prevailing microscopy technique employed in regulatory biomonitoring practices within France, namely DIC (Differential Interference Contrast) light microscopy [143]. It is noteworthy that no publicly available DIC diatom image datasets existed for training machine learning models. To bridge this gap, we developed methods to extract diatom images from publicly available atlases that were developed by practitioners and generate synthetic datasets, accentuating the originality of this contribution. However, acknowledging the broader community of diatom experts who utilize the brightfield technique [143], we established a collaborative effort with researchers from the University of Duisburg-Essen (https://www.uni-due.de/phycology/). This collaboration

aimed to broaden the scope of our models by testing them on their dataset, which is set to be made available soon. Moreover, to showcase the versatility of the developed methods across various applications, we conducted evaluations on standard computer vision datasets.

This section describes the datasets that were used in the thesis:

1.2.1 Diatom Datasets

DIC Full microscopy images

For detection, the microscopy images were obtained from two types of diatom samples:

1. **CLEURIE -** These are natural samples collected from the Cleurie Stream, Vosges, France on hard mineral substrates and fixed by adding ethanol, showing natural communities consisting of various diatom species. This dataset has a diverse variety of diatom taxa, with widely varying morphology.

2. **NPAL -** These are samples collected from a lab culture of the diatom *Nitzchia palea* (NPAL). The images contain a single diatom taxon, and hence the morphology is not as diverse as the CLEURIE dataset, but these individuals could exhibit different in-class morphological variances.

Each liquid sample was subjected to oxidation using hot hydrogen peroxide in order to remove organic material, rinsed with deionized water, then mounted on permanent microscope slides with Naphraxinto, a high refractive index resin in order to be visualized at 1000× magnification using light microscopy. Figure. 1.5 shows examples of images from the CLEURIE and NPAL datasets.

Instead of manually extracting the different morphological parameters from each diatom, the detection tool could aid the biologists in automatically detecting and extracting the parameters.

DIC Individual Images - Atlas Dataset

The classification network takes individual diatom instances and predicts the corresponding taxon. The Atlas dataset consists of diatom images extracted from open-access diatom atlases belonging to five different regions: Rhône-Alpes [13], Île-de-France [96], Bourgogne [139], Loire [48], and Provence-Alpes-Côte d'Azur [47]. From the atlases in PDF format, the diatom images were automatically extracted using PDF scraping. This is a cheap and effective way to obtain taxonomy-based classification and detection dataset, since the individual diatom images are freely available, along with their taxonomy for ground

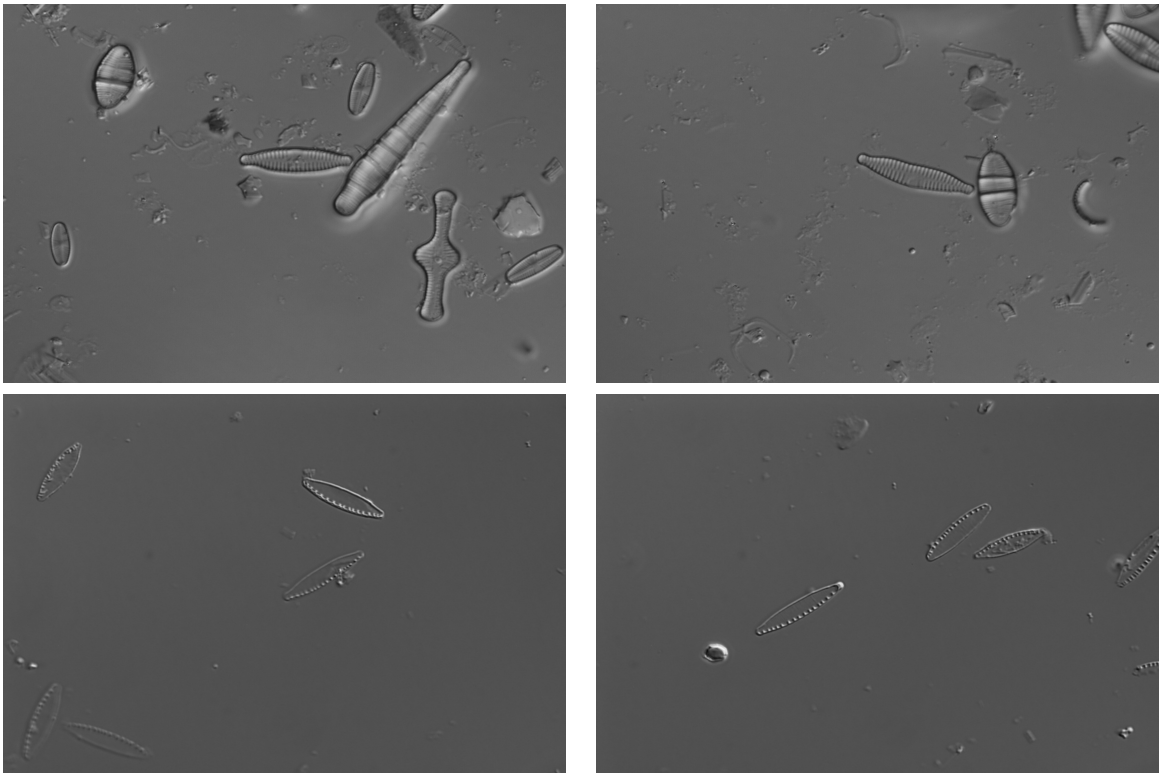


Figure 1.5: Example of datasets used for detection: CLEURIE (top) and NPAL (bottom). The dataset contains images acquired from two different kinds of samples. CLEURIE is obtained from natural samples, and hence contains a diverse variety of diatom taxa. NPAL dataset consists only of the diatom taxon *Nitzchia palea*.

truth labels. For training purposes, we focused on taxa that had a minimum of 30 diatom images available. As a result, the dataset was composed of 197 diatom taxa, encompassing a substantial collection of 14,694 images. These images were resized to a uniform size of 256x256 pixels. Moreover, to ensure the accuracy of the taxonomy within the dataset, expert diatomists meticulously verified and validated the taxonomic classifications. Figure 1.6 shows the distribution of the number of images per taxon. It follows a long-tail distribution, with the maximum number of images per taxon is 504 and the minimum number of images per taxon is 30. Some images extracted from the atlases are shown in Figure. 1.1. The list of available species and the number of images per species is provided in Table A.1.

Debris

The **Debris dataset** consists of 600 images of a diverse variety of debris objects, extracted from microscopy images. Together, the atlas and the debris dataset are used to generate synthetic microscopy images, explained in Section 3.3.

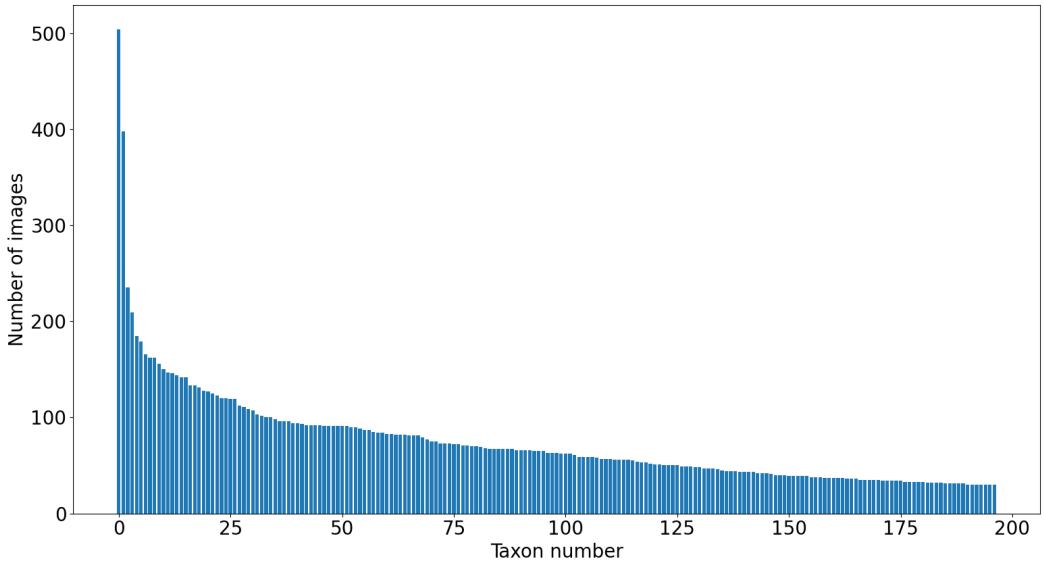


Figure 1.6: Distribution of number of images for every taxon in the Atlas dataset. The distribution is long-tailed, with a maximum of 504 images and a minimum of 30 images per taxon. The list of available species and the number of images per species is provided in Table A.1.

Brightfield images - The UDE Diatom dataset

This dataset was obtained in collaboration with Dr. Bánk Beszteri and Dr. Michael Kloster from University of Duisburg-Essen, Germany. 320 benthic samples from 14 different river/lake ecotypes and 1 multi-stressor experiment were collected. The slide specimens were converted into digital format using a VS200 slide scanning microscope in bright-field mode, utilizing a UPLXAPO60XO 60x/1.42 oil immersion objective. The dataset consists of 83,354 light microscopy images of individual diatoms. We consider two subsets of the dataset for the experiments: (i) D25: this dataset includes taxa with number of samples ≥ 25, consisting of 238 taxa (ii) D50: this dataset includes taxa with number of samples ≥ 50, consisting of 176 taxa.

1.2.2 Other Datasets

The following datasets were used to evaluate the developed models: WHOI-Plankton [133], CIFAR10 [93], CIFAR100 [93], FashionMNIST [193], SVHN [130] and CUB-200-2011 [186].

1.2.3 Dataset annotation

Obtaining annotated dataset is one of the key requirements for training a deep learning model. For most of our labelling tasks, we relied on the labelling platform Labelbox [3] with Model Assisted Labelling. Labelbox was used for annotating the axis-aligned bounding boxes and semantic segmentation masks. The semantic segmentation masks for the NPAL dataset were annotated by Clara Martinez and Sarah Chéron. For labelling rotated bounding boxes, we used roLabelImg [1].

1.3 Overview of the thesis

The aim of this thesis is to develop an end-to-end pipeline for automatic diatom detection, classification and morphometric analyses that can be deployed in a real-world system. To achieve this, it is important to ensure that the different blocks of the pipeline are robust, and the predictions obtained from them are reliable. The methods developed in this thesis aim to address these challenges. However, there are several obstacles that must be overcome to achieve this goal. First, the quality and appearance of the images change drastically depending on the type of microscope used or focus level, and it is impractical to develop specialized methods for every image type. In other words, the tools must be easily generalizable to different types or quality of images and should be easily usable without much of manual intervention or the need for tuning different parameters. Second, it is important that the users have a reliable estimate of the confidence that they can have on the prediction obtained from the system. Thus, in accordance with these requirements, we will primarily focus on the development of robust feature extraction and classification methods. We will focus on methods for uncertainty estimation that will provide users with an accurate estimate of the confidence level of the system's predictions.

The thesis is organized as follows:

- **Chapter 2** reviews the different building blocks of our end-to-end pipeline for automatic diatom detection and classification. Specifically, the chapter will focus on the various detection and classification methods that were employed in our pipeline. We will begin by introducing traditional hand-crafted methods that have been used for detection and classification. To overcome their limitations, we introduce state-of-the-art approaches that utilize deep learning methods. These deep learning models enhance detection and classification performance significantly, offering a robust alternative to traditional techniques. Furthermore, the significance of uncertainty estimation in deep learning-based approaches is discussed, along with various methods for estimating uncertainty. These

methods provide users with reliable confidence estimates for the system's predictions.

- In **Chapter 3**, we focus on diatom detection methods. We will discuss the state-of-the-art approaches used in automatic diatom detection. One of the primary challenges faced in training deep learning models is the lack of a sufficient amount of labelled data. To address this issue, we propose a method for generating synthetic datasets that can be used to augment the limited labelled datasets. We demonstrate how this synthetic dataset approach can significantly boost the performance of detection methods, when only limited labelled datasets are available. Finally, a sensitivity analysis is performed to study the impact of different parameters used for generating the synthetic images.

  In addition, we also present a real-life application of our automatic detection tools, that involves instance segmentation and morphological parameter extraction from the obtained segmentation masks. This will be further used by biologists for their morphometric analyses.

- In **Chapter 4**, we study the classification methods for diatoms and uncertainty estimation. We will discuss the state-of-the-art approaches that have been developed for this task, as well as the challenges in diatom classification. One of the primary challenges in diatom classification is the high inter-class similarity and intra-class variance of the different diatom species. This can make it difficult for traditional classification methods to accurately distinguish between different species. To address this issue, we will propose a method for learning feature representations that can group visually similar-looking images of each class together, while also ensuring that the inter-class features are widely separated from each other.

  Additionally, we introduce a method for estimating uncertainty in classification performance. This method involves using the proximity of a data point to different class features to estimate the uncertainty in the network's prediction. We also show how this method can be used to obtain a reliable estimate of the prediction confidence and detect out-of-distribution samples. Finally, we demonstrate the effectiveness and generalizability of our proposed feature representation and uncertainty estimation method by testing on various standard datasets.

- The application of the automatic diatom identification pipeline for Biotic Diatom Index (BDI) calculation is the focus of **Chapter 5**. The BDI calculation is based on the identification and enumeration of different diatom taxa present in a benthic sample. However, misclassifications can occur during the identification process, which can lead to errors in the BDI calculation. In this chapter, we analyse the uncertainty in BDI

values due to misclassifications. To do this, we generate synthetic diatom observations and introduce misclassifications in these observations to study how it impacts the BDI error. We also analyse the robustness of the classification network to input perturbations. Finally, we also combine the automatic diatom identification pipeline with hierarchical classification, since identification at higher taxonomic levels could reduce the misclassification errors by giving the users a set of potential classes that they can search for.

- In **Chapter 6**, we focus on the topic of image retrieval. Image retrieval is the task of searching for images that are similar to a query image based on their visual content. It can be a useful tool for biologists to quickly and easily find similar images of diatoms for comparison and identification purposes. Diatom identification often requires comparing microscopic images of different diatom species and subspecies, which can be a time-consuming and challenging task, particularly for non-experts. By using an image retrieval system, researchers can input an image of a diatom they are interested in identifying, and the system will retrieve similar images of diatoms from a large dataset based on their visual content. This can help to quickly narrow down potential matches and provide a starting point for further investigation. In this chapter, we propose a novel method for generating a hierarchy of images based on their visual appearance. We develop a query search method that combines the hierarchical structure with traditional distance based method for efficient and effective image retrieval, allowing users to find the most relevant images for their needs. To demonstrate the effectiveness of our proposed method, we compare it to other state-of-the-art image retrieval methods on standard fine-grained visual datasets.

- **Chapter 7** concludes the thesis and provides discussions on the limitations and future work.

1.4 Contributions

Publications in international journals and conferences:

- [182] **Aishwarya Venkataramanan**, Martin Laviale, and Cédric Pradalier. "Integrating visual and semantic similarity using hierarchies for image retrieval". *In International Conference on Computer Vision Systems*, pages 422–431. Springer, 2023. (Chapter 6)

- [179] **Aishwarya Venkataramanan**, Assia Benbihi, Martin Laviale, and Cédric Pradalier. "Gaussian Latent Representations for Uncertainty Estimation using Ma-

halanobis Distance in Deep Classifiers". *In Proceedings of International Conference on Computer Vision Workshop*, pages 4488–4497, 2023. (Chapter 4)

- [180] **Aishwarya Venkataramanan**, Pierre Faure-Giovagnoli, Cyril Regan, David Heudre, Cécile Figus, Philippe Usseglio- Polatera, Cédric Pradalier, and Martin Laviale. "Usefulness of synthetic datasets for diatom automatic detection using a deep-learning approach". *Engineering Applications of Artificial Intelligence*, 117:105594, 2023. (Chapter 3)

- [181] **Aishwarya Venkataramanan**, Martin Laviale, Cécile Figus, Philippe Usseglio-Polatera, and Cédric Pradalier. "Tackling inter-class similarity and intra-class variance for microscopic image-based classification". *In International Conference on Computer Vision Systems*, pages 93–103. Springer, 2021. (Chapter 4)

Under preparation for submission in journals:

- Sarah Chéron, Vincent Felten, **Aishwarya Venkataramanan**, Carlos Eduardo Wetzel, David Heudre, Cédric Pradalier, Philippe Usseglio-Polatera, Simon Devin, MArtin Laviale. "Comparative effects of glyphosate and AMPA on the growth and morphology of common freshwater benthic diatoms". (Chapter 3)

- **Aishwarya Venkataramanan**, Michael Kloster, Andrea Burfeid Castellanos, Mimoza Dani, Serge Mayombo, Danijela Vidakovic, Daniel Langenkämper, Mingkun Tan, Cédric Pradalier, Tim Nattkemper, Martin Laviale, Bánk Beszteri. "Addressing key challenges of digital light microscopic diatom analysis by a freshwater image data set and deep learning models". (Chapter 4)

- **Aishwarya Venkataramanan**, Philippe Usseglio-Polatera, David Heudre, Cédric Pradalier, Martin Laviale. "Error propagation in the Biological Diatom Index (BDI) due to taxonomic misclassification based on a deep-learning approach". (Chapter 5)

Poster presentations at international conferences:

- Michael Kloster, Andrea Burfeid-Castellanos, Danijela Vidacović, Ntambwe Albert Serge Mayombo, Mimoza Dani, **Aishwarya Venkataramanan**, Cédric Pradalier, Martin Laviale, Bánk Beszteri. "The new University of Duisburg-Essen diatom digital image training data set (UDE did it v1.0)". *16th International Diatom Symposium*, Aug 2023, Yamagata, Japan

- Sarah Chéron, Vincent Felten, David Heudre, **Aishwarya Venkataramanan**, Simon Devin, Martin Laviale. "Link between glyphosate exposure and teratological forms in

freshwater benthic diatoms". *SETAC Europe 32nd annual meeting*, May 2022, Copenhagen, Denmark.

Oral and poster presentations at national conferences:

- **Aishwarya Venkataramanan**, Philippe Usseglio-Polatera, David Heudre, Cédric Pradalier, Martin Laviale. "Reconnaissance automatique des diatomées: évaluation de l'influence des erreurs d'identification sur l'IBD et de l'état écologique des cours d'eau. Exemple du bassin Rhin-Meuse". *41ème colloque de l'Association des Diatomistes de Langue Française (ADLaF)*, 12-14 September 2023, Besançon, France (Oral)

- Sarah Chéron, **Aishwarya Venkataramanan**, Clara Martinez, Huyên Phong Hamon, David Heudre, Cédric Pradalier, Philippe Usseglio-Polatera, Simon Devin, Vincent Felten, Martin Laviale. "Effet des pesticides sur la morphologie des diatomées benthiques d'eau douce : le cas du glyphosate". *Colloque annuel de la Société d'Écotoxicologie Fondamentale et Appliquée (SEFA 2022)*, June 2022, Metz, France (Poster)

- Pierre Faure-Giovagnoli, **Aishwarya Venkataramanan**, David Heudre, Thibault de Garidel-Thoron, Philippe Usseglio-Polatera, Cédric Pradalier, Martin Laviale. "Utilisation d'images microscopiques artificielles en complément d'images réelles pour la détection automatique de diatomées par apprentissage profond". *Deep Learning pour le traitement et l'analyse d'image et de son en écologie*, 16 November 2020, Lyon, France (Oral)

- Pierre Faure-Giovagnoli, **Aishwarya Venkataramanan**, Cécile Figus, David Heudre, Thibault de Garidel-Thoron, Camille Noûs, Philippe Usseglio-Polatera, Cédric Pradalier, Martin Laviale. "Classification automatique des diatomées par apprentissage profond pour l'amélioration du diagnostic écologique des milieux aquatiques". *5ème colloque biennal des Zones Ateliers-CNRS*, 3 November 2020, Blois, France (Oral)

Chapter 2

Concepts and Related Works

This chapter reviews the concepts relevant to the thesis. Section 2.1 provides an introduction to the different types of learning algorithms. We will be using deep learning, particularly Convolutional Neural Network (CNN) for most of the tasks, hence, we provide a primer on the CNN concepts in Section 2.2. The subsequent sections focus on the different deep learning architectures and methods that will be used. Section 2.4 discusses the principle behind bounding box detection. Section 2.5 explains the working of semantic segmentation. The different classification methods are explained in Section 2.3. Finally, Section 2.6 focuses on hierarchies, which will be used in image retrieval and in BDI calculation.

While this chapter mainly reviews the working principle behind the different deep learning methods used in the thesis, each chapter has a dedicated literature survey section on the methods that have been applied to diatoms or other relevant datasets.

2.1 A Brief Introduction to Learning Algorithms

Machine learning is a branch of artificial intelligence (AI) that involves training algorithms to learn patterns in data and make predictions or decisions without being explicitly programmed to do so. Deep learning is a subfield of machine learning that involves the use of artificial neural networks to learn and make predictions on data. These neural networks are composed of multiple layers of interconnected nodes, which allow them to learn complex representations of data by progressively extracting more abstract features from the input data. The algorithms can be broadly divided into three categories:

- **Supervised Learning -** This type of algorithm learns from labelled data, where the inputs and the desired outputs are provided. The inputs can be images, videos, text, hand-crafted features, etc. The algorithm tries to learn a function that maps the inputs

to the outputs by minimizing the error between its predicted outputs and the true labels. There are two main types of supervised learning algorithms:

  - Regression: In regression, the output label is a continuous variable, and the goal is to predict a numeric value for the output. One example of regression algorithm that is used in the thesis is bounding box detection, where the network predicts the bounding box coordinates of the diatoms in the input image.
  - Classification: In classification, the output label is a categorical variable, and the goal is to predict a class label for the input features. Two types of classification are employed in this thesis: (i) image segmentation or pixelwise classification, where every pixel in the input image is classified into pre-defined labels, and (ii) image classification – every input image is categorized into predefined labels.

- **Unsupervised Learning -** Unsupervised learning is a type of learning algorithm where the machine learning model is trained on unlabelled data, which consists of input features without corresponding output labels. The goal of unsupervised learning is to find patterns, structure, or relationships in the data.

  There are two main types of unsupervised learning which will be used in the thesis:

  - Clustering: In clustering, the goal is to group similar data points together into clusters, based on their similarity in terms of features or other measures. Some examples of clustering methods include K-Means, DBSCAN, X-Means, Gaussian Mixture Models and hierarchical clustering.
  - Dimensionality Reduction: In dimensionality reduction, the goal is to reduce the number of features in the data while preserving as much of the relevant information as possible. This can help to simplify the data and make it easier to analyse. Some well known dimensionality reduction methods include the Principal Component Analysis (PCA), t-SNE [177], Linear Discriminant Analysis (LDA).

- **Reinforcement Learning -** Reinforcement learning is a type of learning algorithm where an agent interacts with an environment and learns to take actions that maximize a cumulative reward signal over time. The agent receives feedback in the form of rewards or penalties based on the actions it takes in the environment. Reinforcement learning is often used in applications such as game playing, robotics, and autonomous systems. This type of learning is not used in the thesis.

Another type of algorithm that is gaining popularity in the deep learning literature is self-supervised learning. Self-supervised learning leverages the inherent structure or patterns

present in the data to generate labels or tasks for training the model. In this work, we use a combination of supervised, unsupervised and self-supervised learning for classification and uncertainty quantification.

2.1.1 Some jargons in machine learning

- Loss function – A mathematical function that measures how well a model is able to predict the expected output for a given input. It is a quantification of the difference between the predicted output and the actual output.
- Overfitting – A condition in which a machine learning model is trained too well on the training data and fails to generalize to new data.
- Underfitting – A condition where a model is unable to capture the underlying patterns and relationships in the training data, resulting in poor performance on both the training and test data.
- Regularization – A technique used to prevent overfitting in machine learning models by modifying the learning algorithm, typically by adding a penalty term to the loss function.
- Optimizer – An optimizer is an algorithm or method used to adjust the parameters of a model to minimize the difference between predicted and actual outcomes during training. The goal of an optimizer is to find the optimal set of parameters that produce the best performance of the model on a given task.
- Latent space – An abstract multi-dimensional space containing feature values such that similar data points are closer together in space.
- Batch - A batch refers to a subset of the data that is processed together by the network.

2.2 Deep Learning Concepts

A Neural Network (NN) is a collection of interconnected nodes or neurons designed to process and manipulate input data. In the context of a Deep Neural Network (DNN), this architecture comprises numerous layers of interconnected neurons situated between the input and output layers. These are also known as fully connected layers or dense layers and perform a linear combination of the features from the preceding layers. A Convolutional Neural Network (CNN), which is extensively used in the thesis, is different, where the network is

made of convolutional layers. The input is passed through these layers, which hierarchically extracts the different features and learns complex patterns in the data.

2.2.1 Model Components

The fully connected layer and the convolution layer are two of the most commonly used layers of neural networks. While the fully connected layer connects every neuron in one layer to every neuron in the next layer, the convolutional layer connects neurons to a grid-structures inputs such as images or videos. Other types of layers include pooling layers, normalization layers and activation layers. This section reviews these layer types.

Fully Connected Layer (Dense Layer)

A fully connected layer (also known as a dense layer) is the simplest layer present in a neural network. This layer applies linear transformation to the input vector. Mathematically, a fully connected layer can be represented as a matrix multiplication between the input vector x and a weight matrix W, followed by the addition of a bias vector b, and the application of an activation function σ:

$$y = \sigma(W^T x + b) \tag{2.1}$$

where $x \in \mathbb{R}^M$ is the input vector, $W \in \mathbb{R}^{N \times M}$ is a weight matrix, $b \in \mathbb{R}^N$ is a bias vector, $y \in \mathbb{R}^N$ is the output vector, and σ is the activation function applied element-wise to the resulting vector.

Convolution Layer

A convolution layer is a type of layer in a neural network that performs a mathematical operation called convolution on the input. The convolution operation involves sliding a small, fixed-sized window called a kernel or filter over the input, computing the dot product between the kernel and the input at each location, and producing a new output feature map. The kernel is typically learned during training and represents a set of weights that determine the features the convolution layer will extract.

Let $x \in \mathbb{R}^{H \times W \times C}$ be the input, where H is the height, W is the width, and C is the number of channels. Let $k \in \mathbb{R}^{N \times M \times F}$ be the kernel or filter, where N and M are the kernel size and F is the number of filters. Let $b \in \mathbb{R}^F$ be the bias.

Then, the output feature map $y \in \mathbb{R}^{H \times W \times F}$ of a convolutional layer with activation function σ is defined as:

$$y_{i,j,f} = \sigma\left(\sum_{c=1}^{C}\sum_{n=1}^{N}\sum_{m=1}^{M} k_{n,m,f}x_{i-n,j-m,c} + b_f\right) \tag{2.2}$$

where i and j are the spatial indices of the output feature map, and f is the filter index. The activation function σ is applied element-wise to the resulting feature map. To prevent border effects that reduce the size of the output by half of the kernel size, one can use padding. Padding is the process of adding extra rows and columns of zeros to the input feature map, which allows the kernel to move over the entire image and generate an output feature map of the same size as the input.

Pooling Layer

Pooling layers are a type of layer in a neural network that downsample the input feature maps by summarizing local neighbourhoods of activations. Pooling is usually applied after the convolutional layer in CNNs to reduce the spatial dimensions of the feature maps and to extract the most important information from them. There are two widely used pooling layers: max pooling and average pooling. In max pooling, the output value of each pooling unit is the maximum value in a local neighbourhood of the input feature map, whereas in average pooling, the output value is the average value of the same neighbourhood.

Mathematically, a max pooling operation can be represented as follows:

$$y_{i,j,c} = \max(x_{i:i+m,j:j+n,c})$$

where y is the output feature map, x is the input feature map, and c is the channel index. The max pooling operation takes the maximum value over a local window of size $m \times n$, and (i, j) is the position in the input feature map along each channel.

Similarly, an average pooling operation can be represented as follows:

$$y_{i,j,c} = \text{mean}(x_{i:i+m,j:j+n,c})$$

Pooling layers can help to reduce the size of the feature maps and to control overfitting by introducing a form of regularization. They can also make the network more robust to small translations in the input, as the pooling operation will typically produce the same output regardless of the exact location of the feature in the input.

Normalization Layer

Normalization layers are a type of layer used in neural networks to normalize the input data in order to make it easier to learn from. The purpose of normalization layers is to transform

the input data so that it has a certain statistical property, such as zero mean and unit variance, that makes it easier for the network to learn from.

There are several types of normalization layers, including batch normalization, layer normalization, and instance normalization. Batch normalization is the most commonly used type of normalization layer and is applied after a convolutional or fully connected layer in a neural network.

In batch normalization, the input data is normalized to have zero mean and unit variance across the batch dimension. This is done by subtracting the mean and dividing by the standard deviation of the input data for each channel.

Activation Functions

Activation functions are mathematical functions applied to the output of a neural network layer to introduce nonlinearity into the model. Without activation functions, a neural network would be limited to linear transformations of the input data, making it unable to learn complex patterns and relationships in the data. Some commonly used activation functions are the sigmoid, hyperbolic tangent, softmax, ReLU and Leaky ReLU.

Softmax. Softmax is generally applied to the final layer of a CNN classifier. It maps input values to a probability distribution over the possible output classes, ensuring that the sum of the probabilities is 1. For instance, if there are N output classes, the final layer consists of N neurons. Let the final layer be represented as $\boldsymbol{z}$. Then the individual neurons are represented as $\{z_1, z_2, ..z_N\}$. The softmax value of every neuron is given by

$$\sigma(z_i) = \frac{e^{z_i}}{\sum_{k=1}^{N} e^{z_k}} \tag{2.3}$$

Sigmoid. Similar to softmax, sigmoid is another activation function that is applied to the final layer of the CNN classifier. The sigmoid function maps any input value to a value between 0 and 1, making it useful for binary classification problems. The sigmoid function is given by:

$$\sigma(z_i) = \frac{1}{1 + e^{z_i}} \tag{2.4}$$

Hyperbolic Tangent (Tanh). The tanh function is similar to the sigmoid function, but maps input values to a range between -1 and 1. It can be useful for problems where the output should be balanced around zero. The tanh is given by:

$$\sigma(z_i) = \frac{e^{z_i} - e^{-z_i}}{e^{z_i} + e^{-z_i}} \tag{2.5}$$

Rectified Linear Unit (ReLU). The ReLU function sets all negative input values to zero, and leaves positive values unchanged. It is the most widely used activation function due to its simplicity and effectiveness. The ReLU is given by:

$$\sigma(z_i) = \max(0, z_i) \tag{2.6}$$

Leaky ReLU. The Leaky ReLU function is similar to ReLU, but allows a small, non-zero gradient c for negative input values. The leaky ReLU is given by

$$\sigma(z_i) = \begin{cases} z_i, & \text{if } z_i \geq 0, \\ cz_i, & \text{if } z_i < 0. \end{cases} \tag{2.7}$$

There are several other activation functions available, depending on the type of problem to solve. A comprehensive survey on these functions is available in [45].

Dropout

Dropout is a regularization technique used in neural networks to prevent overfitting. It works by randomly setting a fraction of the output features of a layer to zero during training. This means that the network can't rely on any single feature or neuron to make predictions, and must instead learn more robust representations of the data. The fraction of neurons to be dropped is determined by a hyperparameter, typically between 0.2 and 0.5. During inference (i.e., when making predictions on new data), dropout is turned off, and all neurons are used.

The coming sections discuss the applications of CNNs in computer vision tasks.

2.3 Classification

A classification network takes an input and categorizes it into a set of predefined labels. As with the detection methods, classifiers can be divided into deep learning based classifiers and the non-deep learning based ones. Each method has its own advantages and disadvantages. The non-deep learning methods typically use hand-crafted features. Some classic hand-crafted feature based classifiers include Naive Bayes, Decision Trees, Random Forests, Support Vector Machines, K-Nearest Neighbours. Refer to [69] for a detailed explanation of each of these methods. These methods require fewer computational resources over the deep learning based ones, and are interpretable. Deep learning methods typically use several layers of neural networks or CNNs to automatically extract the features and have shown remarkable performance over the traditional methods. The advantage over the traditional

classifiers are their ability to learn complex representations, robustness to noisy data and the ability to be trained end-to-end, without additional requirement for feature engineering, which makes them an attractive choice when dealing with complex inputs such as images.

A deep learning classification network relies on two main components: feature extractor and classifier. The feature extractor is generally a CNN that takes high dimensional inputs such as images and generates low dimensional representations. The CNN extracts the relevant features needed to represent the input data. The classifier uses the representations from the CNN to assign probability values for each class and decide the classification labels. The general working is described below:

- **Input and Feature Extraction -** The data that needs to be classified is given as input to a DNN. The neural network acts as an automatic feature extractor of the input. During training, it learns to extract the relevant features needed for classifying the data. Deep neural networks consist of several layers of fully connected or convolution layers, which captures hierarchical representations of the data. With increasing depth, the network captures increasingly complex features needed to effectively classify the data. For instance, in a CNN with an image as the input, the initial layers extract some basic features such as edges, corners, and colour blobs. At deeper levels, the network extracts features that capture larger structures and patterns in the image, such as object parts, regions, texture, object classes or scene categories. The pooling and activation layers capture the important features and the non-linearity present in the data respectively.

- **Classification -** The classification layer is usually implemented as one or more fully connected layers (also known as dense layers) followed by an activation function.

  The final output of the classification layer is typically passed through a softmax activation function, which converts the output into a probability distribution over the different classes or categories. Finally, the class with the highest probability is predicted as the class label for the input data.

Classification networks are generally optimized using the cross-entropy loss. The probability values obtained from the softmax are used to estimate the cross-entropy loss. The loss value increases proportional to the difference between the predicted probability and the actual probability (typically 1) of the ground truth class. It is given by:

$$\mathcal{L}_{\text{cross-entropy}} = -\sum_{i=1}^{K} y_i \log(p_i) \tag{2.8}$$

where K is the total number of samples, y_i is the binary one-hot encoding value corresponding

to the target class, for the ground truth class 1, and p_i is the probability predicted by the network.

Once the network is trained, it can be used for making predictions on new, unseen data by passing the data through the feature extraction layers of the CNN followed by the trained classification layer.

One of the earliest CNNs for classification is LeNet-5 [101]. This architecture was designed in the late 90s for handwritten number classification and had fewer number of layers compared to the modern architectures, making it suitable for small datasets. The biggest breakthrough in the deep learning based classification literature was due to two reasons: the ImageNet [38] dataset and the AlexNet architecture [94]. In 2009, one of the first large scale labelled datasets, ImageNet, was introduced for visual recognition. The widely used ImageNet-1k consists of over 1 million labelled images of commonly observed objects, categorized into 1000 classes. 2012 marked the breakthrough in the deep learning field, when AlexNet was introduced for classification, consisting of multiple convolutional and pooling layers, followed by fully connected layers, and also incorporated the use of ReLU activation function [54] and dropout regularization [166]. AlexNet was one of the first CNNs to show the effectiveness of deep learning in large-scale image classification tasks. The model architectures have since evolved and continue to do so in a rapid pace, and have drastically brought down the error rates in classification. Some well-known architectures include VGG [164], Inception [167], ResNet [71], DenseNet [77], MobileNet [74], and EfficientNet [169]. VGG is a deep CNN architecture known for its simplicity and effectiveness in image classification tasks. The network consists of multiple convolutional layers, followed by max pooling layers, and ends with fully connected layers for classification. Inception was designed to improve the efficiency by reducing the number of parameters without sacrificing accuracy. It uses a combination of 1x1, 3x3, and 5x5 convolutions in parallel to capture features at different scales. As the number of layers in CNNs increased, they suffered from the vanishing gradient problem. The vanishing gradient problem is a phenomenon where the gradients of the loss function become very small or vanish during the training of deep neural networks. This can cause the network to fail to learn important features in the data, and its performance may suffer. ResNet addresses this problem by introducing residual connections that allow gradients to propagate more easily. The network consists of multiple residual blocks that contain convolutional layers, batch normalization layers, and shortcut connections. DenseNet connects each layer to every other layer in a feed-forward fashion to maximize feature reuse and improve the flow of information through the network. The network consists of multiple dense blocks that contain convolutional layers and concatenation operations. MobileNet is designed to be computationally efficient and well-suited for mobile and embedded devices.

It uses depth-wise separable convolutions, which separate the spatial and channel-wise convolutions to reduce the number of parameters and computations. EfficientNet optimizes the trade-off between model size and accuracy using a compound scaling technique. It uses a combination of depth-wise separable convolutions, squeeze-and-excitation blocks, and a compound scaling technique to achieve high accuracy while keeping the model size small.

The current state-of-the-art classification models use Vision Transformers (ViTs) [42] for feature extraction, that uses self-attention mechanisms to capture long-range dependencies in images. While ViTs have shown excellent performances on large scale datasets, they tend to overfit when presented with fewer data. Current research focuses on creating data efficient ViTs [172]. In this work, since we deal with small scale datasets, we use CNNs for training the models.

2.4 Bounding Box Detection

The goal of an object detection network is to localize instances of interest and classify them into a set of predefined object classes. Traditional methods use features extracted from Histogram of Gradients (HOG) [37], Scale-Invariant Feature Transform (SIFT) [117], Speeded-Up Robust Features (SURF) [10] or Haar-like cascades [184]. These methods extract key points or interest points from an image and use them to describe the object's local features. The features can then be matched to a database of known object features for detection. Template matching is a simple method for object detection that involves comparing an image patch (template) with regions of an input image to find areas that match the template [16]. These methods can be effective for object detection in scenarios where objects have distinct visual properties, but may not be robust to variations in scale, rotation, or lighting. In recent years, there has been a significant improvement in object detection models due to the emergence of deep learning. These models have surpassed previous benchmarks using traditional methods and achieved the highest accuracy rates, resulting in impressive performance [80].

Bounding box object detection networks in deep learning are of two broad categories: one-stage and two stage. The one-stage family [110, 113, 148] generally uses a unique network to extract and classify the bounding boxes, while two-stage networks [61, 62, 151] first use a Region Proposal Network to propose multiple Regions of Interest (RoI) for the bounding boxes, which are subsequently classified using a second generic backbone architecture. Due to the additional step in two-stage networks, they are slower, but achieve better performance than the one stage networks.

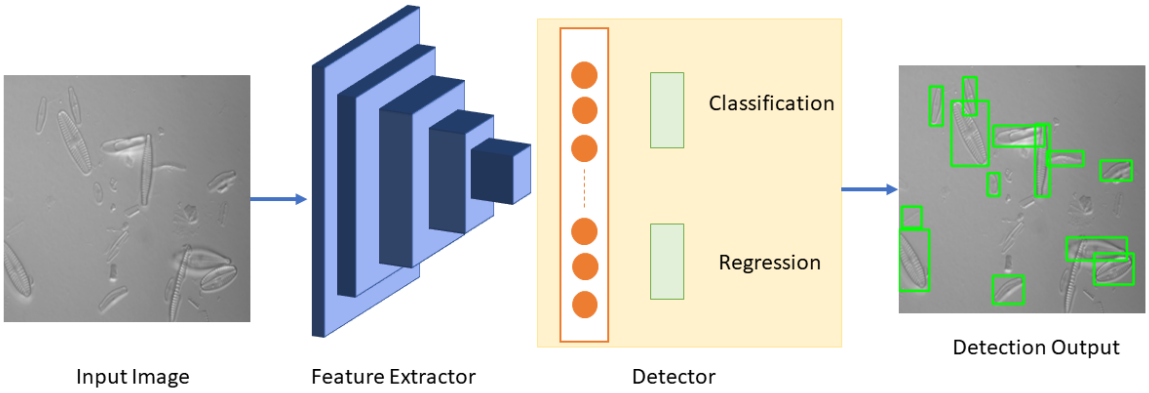


Figure 2.1: Diagram depicting the general pipeline of one stage detection networks.

2.4.1 One Stage Networks

The working of one-stage horizontal box detection networks can be summarized as follows:

1. **Input and Feature Extraction -** The network's input is an image on which detection is to be performed. A CNN backbone acts as a feature extractor and extracts high level features from the image. This CNN backbone is typically a pre-trained classification network, trained on the ImageNet dataset [38].

2. **Object Localization -** The backbone outputs a stacked set of feature maps that represents the original image in a low spatial resolution. The feature maps are used to predict a set of bounding boxes of potential objects in the image. Each bounding box is represented by four values: (c_x, c_y, w, h), where c_x and c_y are the x and y coordinates of the box centres relative to the original image, w is the box width and h is the box height. Apart from these four values, the network also outputs a probability value for each box p_{obj}, called the objectness score, which indicates the likelihood of presence of an object in the box region.

3. **Object Classification -** For each bounding box, the network predicts the class of object it contains. The CNN is connected with fully connected layers, which perform a multi-class classification task to identify the objects present in the image. If the detection involves k classes of objects, the classification network outputs k values, representing the probability of presence of each class. Thus, the network outputs a total of $B(5+k)$ values for each image, where B is the number of bounding box predictions, 4 bounding box representation values, objectness score p_{obj} and k class probabilities.

4. **Non-Maximum Suppression -** Most of the time, the network outputs multiple

box predictions for every detected object, resulting in redundant detections. Many of them can be filtered based on the objectness score, where detections below a certain threshold of p_{obj} are eliminated. However, there still remains few redundant predictions. To remove them, a technique called non-maximum suppression is used, where only the box corresponding to the most confident prediction for each object is retained.

Thus, the final output consists of a set of bounding boxes, each associated with a predicted class label and confidence score.

Figure. 2.1 illustrates the pipeline of one stage detectors. Some examples of widely used single stage detection networks include Single Shot Detector (SSD) [113] and the YOLO family of detectors [148]. The SSD architecture is based on a convolutional neural network (CNN) and uses a feature pyramid network to extract features from multiple levels of the image. SSD takes an input image and generates proposals and predicts object classes and locations in a single forward pass through the network. This makes the algorithm fast and suitable for real-time applications. The YOLO algorithm works by dividing the input image into a grid of cells and predicting bounding boxes and class probabilities for each cell. Each bounding box has associated class probabilities, indicating the probability that the object in the box belongs to a particular class. Over the years, several versions of YOLO were released, each performing better than the previous version. The improvements include better quality of features extracted, robustness to variations in object scales and sizes, and improvement in the speed and performance of detection. Single-stage object detection networks are known for their simplicity and speed, thus finding widespread applications in real-time systems. In this thesis, considering the robustness, speed and performance, we use YOLOv5 [81], which was also the state-of-the-art object detection method at the time of experimentation.

2.4.2 Two Stage Networks

Two stage networks consists of a decoupled Region Proposal Network (RPN) and the Object Detection Network. The working can be summarized as follows:

1. **Input and Feature Extraction -** Similar to the one-stage network, the input consists of images, and a CNN extracts the features from the image.

2. **Region Proposal Network -** The Region Proposal Network (RPN) is responsible for generating a set of candidate object proposals in an image. It produces a set of bounding boxes that potentially contain objects. This is done by applying a set of convolutional filters to the input image, which produces a set of feature maps. The RPN then applies a sliding window over these feature maps to generate a set of object

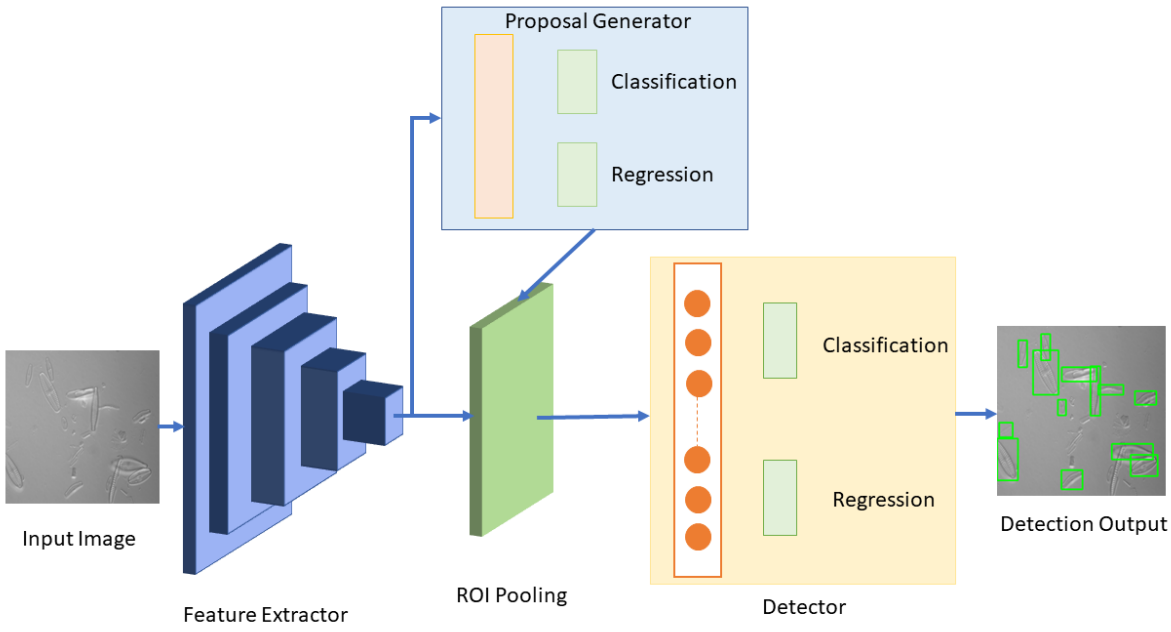


Figure 2.2: Diagram depicting the general pipeline of a two stage detection networks

proposals. The sliding window is a small rectangular region that moves over the feature maps, producing a set of anchor boxes at each position.

The RPN generates numerous object proposals, typically several thousands, and passes them to the next stage of the network.

3. **Object Detection Network -** The part of the network takes the object proposals generated by the RPN and performs object detection. This stage uses a separate convolutional neural network to classify the objects and refine their bounding boxes.

   The object detection network takes the proposals generated by the RPN as input and classifies the objects and refines their bounding boxes.

   The classification step involves predicting the probability that an object is present in each proposal, and the probability that it belongs to each class. The network uses a softmax function to produce a probability distribution over the classes.

   The bounding box regression step involves predicting the offsets between the proposal box and the ground truth box. The network predicts the relative distances between the center of the proposal box and the center of the ground truth box, and the logarithm of the width and height ratios between the two boxes.

   The object detection network applies non-maximum suppression (NMS) to remove du-

plicate detections and selects the highest-scoring detections as the final output.

Since, two stage networks use two separate CNNs for detection, they are slower than the one stage networks. However, having two separate networks allows for better quality feature extraction, and hence more accurate detection. Examples of two stage detection networks include RCNN [62], Fast RCNN [61], Faster RCNN [151]. RCNN performs selective search to generate region proposals, extracts features from each proposal using a CNN, and then uses these features to classify objects and predict their bounding box coordinates. Instead of processing region proposals independently, Fast RCNN shares the feature extraction process among all region proposals, using an ROI (Region of Interest) pooling layer to extract fixed-size feature representations from region proposals. This shared feature extraction process makes Fast RCNN faster during both training and testing compared to RCNN. Faster RCNN uses a Region Proposal Network (RPN) to generate region proposals directly from the CNN features, eliminating the need for an external region proposal generation step. The RPN and the subsequent object classification/bounding box regression network (often implemented as Fast RCNN) share the same CNN features, making the entire process more efficient and faster than RCNN and Fast RCNN.

2.4.3 Rotated Bounding box detection

Most of the bounding box detection methods developed in the computer vision literature deal with axis-aligned bounding boxes, where a rectangular box tightly encloses an object of interest. However, in overhead imagery applications, objects may be oriented at arbitrary angles, which means that the traditional axis-aligned bounding boxes are not the best options to tightly enclose the objects. In such cases, a rotated bounding box can be a more accurate representation. This is also the case with diatom detection, where the diatoms are typically oriented in different angles. A rotated bounding box is a rectangular box that is aligned with the object's orientation, rather than the image's axis. This means that the bounding box can be oriented at any angle, which allows it to more tightly fit the object of interest. In addition to the parameters computed by the axis-aligned bounding boxes, the angle of rotation is also learned by the network. Some methods introduced for rotated bounding boxes are presented in [108, 147, 198, 199].

2.5 Semantic Segmentation

Segmentation networks segment an image into different regions or classes. These networks take an input image and perform pixel-wise classification on the input. The output from

the network is the class predicted by the network for every input pixel. Numerous image segmentation algorithms have been developed in the literature. Early method use techniques such as thresholding [134], region-growing [131], watersheds [129], and active contours [85] for segmenting different objects. A drawback of these methods is that they require manual parameter tuning to obtain good performance. Moreover, the parameters vary with each dataset, adding to the complexity of the process. Learning based methods have proven to be robust over these classical methods and often achieve better performance, motivating their adoption in this work.

A typical deep learning based segmentation network consists of an encoder and a decoder network. The encoder extracts the features from the input image and represents it as a reduced dimensional feature vector. The decoder network uses the extracted features to perform per-pixel classification. The general working of a segmentation network is summarized as below:

1. **Input and Feature Extraction -** The input to the network consists of the image on which segmentation is to be performed. The input is passed through an encoder network, which is made up of several layers of CNN. The encoder consists of a set of convolution layers to learn feature representations of the input image. These features are then downsampled through a series of pooling or strided convolutional layers to reduce the spatial resolution of the feature maps while increasing their depth.

2. **Upsampling -** The decoder is made up of several upsampling layers that gradually increase the spatial resolution of the feature maps while decreasing their depth. The final layer outputs feature maps, whose spatial resolution is the same as the input, and the number of channels equal the number of segmentation labels.

3. **Pixelwise Classification -** The feature maps are passed through a softmax layer. The softmax function translates each element of the feature maps to values between 0 and 1, which represent the probability that the corresponding pixel in the input image belongs to that class. Each pixel in the input image is assigned the class corresponding to the largest probability. Thus, the output of the network is a segmentation mask that has the same spatial resolution as the input image, where each pixel in the mask corresponds to the predicted class label.

One of the early methods using CNN for semantic segmentation is the Fully Convolutional Networks (FCN) [116], which consists of convolutional and pooling layers in the encoder and decoder networks. U-Net [154] uses skip connection between the encoder and the decoder to help preserve the spatial information during the upsampling. SegNet [6] includes a max-pooling index pooling layer in the encoder network that stores the indices of the maximum

values. These indices are then used in the decoder network to perform upsampling and restore the spatial resolution of the feature maps. DeepLab [26, 27, 28] includes a family of networks that use dilated convolutions to increase the receptive field of the network without decreasing the spatial resolution of the feature maps. PSPNet (Pyramid Scene Parsing Network) [205] uses a pyramid pooling module to capture contextual information at different scales. MaskRCNN [70] is an instance segmentation network and is an extension of Faster R-CNN. It includes an additional branch that produces a binary mask for each object instance in the input image. The deep learning literature includes several other types of segmentation networks, and a detailed survey on the different deep learning based segmentation approaches is provided in [127]. Considering the compromise between speed and performance between different networks, we use DeepLabV3 [27] for segmenting the microscopy images.

DeepLabv3 takes an image as input and passes it through a CNN to extract features. The feature extraction process in DeepLabv3 involves using dilated convolutions, which allows the network to capture features at different scales without reducing the resolution of the feature maps. These features are then passed through an atrous spatial pyramid pooling (ASPP) module, which is a set of parallel convolutional layers with different dilation rates. This allows the network to capture context at different scales and improve the accuracy of semantic segmentation. The feature maps obtained from the ASPP module are then passed through a decoder module. The decoder module upsamples the feature maps to the same size as the input image. The upsampled feature maps are then concatenated with the feature maps from the corresponding lower-level layers of the CNN, which provides high-resolution features that help improve the segmentation accuracy. Finally, the concatenated feature maps are passed through a classification layer to predict the class label for each pixel in the input image. The output is a probability map that assigns a probability to each pixel for each class in the dataset.

2.6 Hierarchies for Identification

In machine learning, most of the research on classification focus on flat classification. The classification types explained in the previous section falls in this category, where, given an input, the network predicts the class label that it belongs to. However, in the real-world settings, many of the classification labels are hierarchically related. For example, consider a classification problem involving the following four categories: car, bus, chair, and table. Here, car and bus belongs to the category vehicles, and chair and table belongs to the category furniture. Further, vehicle and furniture belongs to a more abstract category of objects. A hierarchical classification problem takes this relationship into consideration when performing

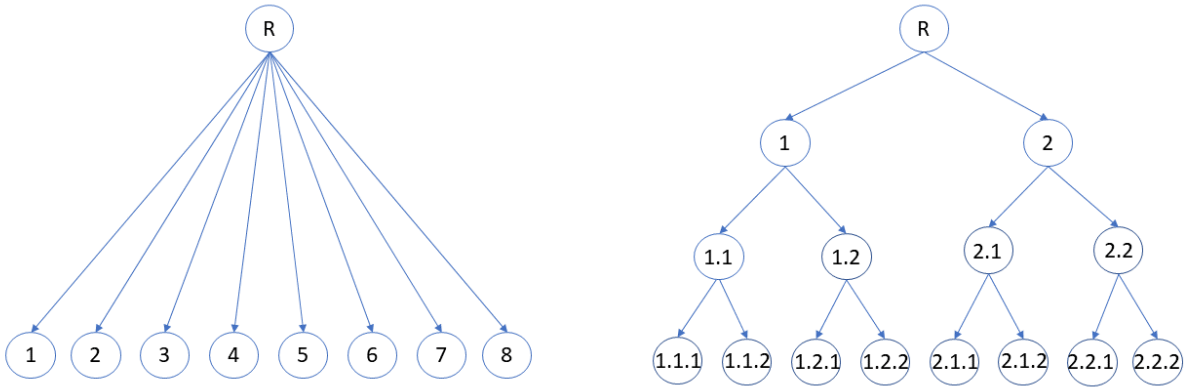


Figure 2.3: (Left) Flat classifier and (Right) Hierarchical Classifier. In a flat classifier, the output classes are independent and there is no relationship or hierarchy between them. A hierarchical classifier organizes the output classes into a tree-like structure, where the classes are arranged in a hierarchy based on some underlying relationships.

classification. The relationship between the different classes are expressed in the form of a tree structure or a Directed Acyclic Graph (DAG), where the most general and abstract classes are at the root of the tree and the most specific and fine-grained classes are at the leaves of the tree. For simplicity, we will only be dealing with tree hierarchy. Figure. 2.3 illustrates the difference between a flat and a hierarchical classifier.

Some terms related to hierarchy:

- Root node – The root node is the highest or topmost node in the tree. It is the starting point for traversing or navigating the tree. In Figure. 2.3, the node 'R' is the root node.
- Parent and child node – A parent node has one or more nodes connected to it as descendants. The descendant is called the child node of the parent. For example, in Figure. 2.3, node 'R' is the parent node for the nodes '1' and '2'. Here, '1' and '2' can be referred to as the children nodes of 'R'. The main difference between a DAG and a tree structure is that a node in DAG can have multiple parent nodes, whereas in a tree, each node has at most 1 parent node.
- Leaf nodes - A leaf node is a node that has no child nodes. It is also referred to as a terminal node. In other words, a leaf node does not have any further branches or descendants below it. In Figure. 2.3, the nodes '1.1.1', '1.1.2', '1.2.1', etc are the leaf nodes.

Hierarchies can be created in various ways, depending on the specific application and domain. Here are two common methods for creating hierarchies:

- Domain Expertise: Hierarchies can be created based on the knowledge and expertise of domain experts. Domain experts can define the categories or classes based on their understanding of the domain, and then organize them into a hierarchy based on their relationships. For example, in a biological context, a domain expert can create a hierarchy of the species under consideration based on their taxonomic classification or morphological characteristics.

- Data-Driven Methods: Hierarchies can also be created in a data-driven manner from the data itself. This can be done through techniques such as clustering. Clustering algorithms can group similar data points together, and the resulting clusters can be organized into a hierarchy based on their relationships. In the context of image classification, the hierarchy could be created based on the visual cues. This approach is also relevant, since classification using CNNs are performed purely based on the visual features extracted from the images.

Hierarchical classification methods have several advantages over traditional flat classification methods, and have been widely used in several application domains [163]. One advantage is that they can improve classification accuracy by taking into account the relationships between classes. Sometimes, the hierarchical relationship between different classes are already known, or can be constructed from lexical databases such as WordNet [126] that organizes words into semantic groups and provides relationships between words. For example, [18] uses WordNet to construct hierarchies and integrates the domain knowledge obtained from it into the traditional classification problem to improve the network's performance. They do so by learning a probabilistic model to compute the likelihood of each node in the hierarchy. [12] extends the traditional cross-entropy loss used in flat classification to work on hierarchies, called the hierarchical cross-entropy loss. To do so, they compute the conditional probability of the nodes at each layer of the hierarchy and calculate the cross-entropy loss at each layer. The sum of them gives the hierarchical cross-entropy loss. HD-CNN [194] learns two sets of classifiers: one for coarse categories and another for fine-tuned classes. Finally, they combine the predictions from both the classifiers to obtain accurate predictions. Another line of work focuses on training a classifier to learn feature embeddings that are representative of the semantic relationships. DeViSE [53] learns a mapping of target classes to a unit hypersphere and using unannotated Wikipedia text to analyse the contexts of related terms. This allows for similar terms to have similar representations. They use a ranking loss function that penalizes outputs that are more similar to incorrect label embeddings than to the correct one. Semantic Embeddings [9] is an embedding algorithm that maps examples onto a hypersphere. The distances on this hypersphere represent similarities that are derived

from the lowest common ancestor height in a given hierarchy tree. They minimized the sum of two different losses: a linear loss based on cosine distance to the embedded class vectors and the standard cross-entropy loss on the output of a fully-connected layer added after the embedding layer.

Apart from improving the classification performance, another advantage of using hierarchies is that they can be more efficient, as the classification process can be stopped once a sufficiently specific class is reached. Additionally, hierarchical classification can be more interpretable, as the hierarchical structure can provide insights into the relationships between different classes.

Chapter 3

Diatom Detection

3.1 Introduction

The microscopy images obtained from the processed samples usually consist of diatoms and debris. The detection step involves localizing the diatom instances in the microscopy images, typically performed manually. Manual localization of the individual diatoms is a laborious and time-consuming task, subject to human biases. Hence, several methods have been introduced to perform the detection automatically since the early 90s [44, 92, 138]. The role of a detector network is to separate the diatom individuals from the occluded or the non-diatom parts. Early methods on detection used hand-crafted methods [19, 91]. Even though these methods achieved reasonable performance, the detection was not fully automatic since these methods required several parameters to be fine-tuned for each dataset. Moreover, they were not robust, and often detected nearby objects, requiring frequent human intervention to filter out the incorrect detections. This motivated the need for methods that are fully autonomous and easily generalizable to different datasets.

Recently, with the introduction of deep learning, tremendous improvements have been made in the domain of object detection. Deep learning methods automatically learn the necessary features needed to localize and differentiate the objects. When provided with sufficient examples, the network can robustly identify the objects of interest. This makes them an attractive alternative over the traditional hand-crafted methods, and are finding wide-spread applications in different domains such as ecology [41, 168], medical image analysis [88], autonomous driving [7] and robotics [65]. However, automatic diatom detection poses several challenges. It is hard to discriminate the individuals from the background of the image due to lack of contrast. This commonly occurs when the diatoms are not in focus when obtaining the images. Moreover, the diatoms frequently overlap with debris and other diatoms, making it hard to isolate the individuals. Figure. 3.1 illustrates this.

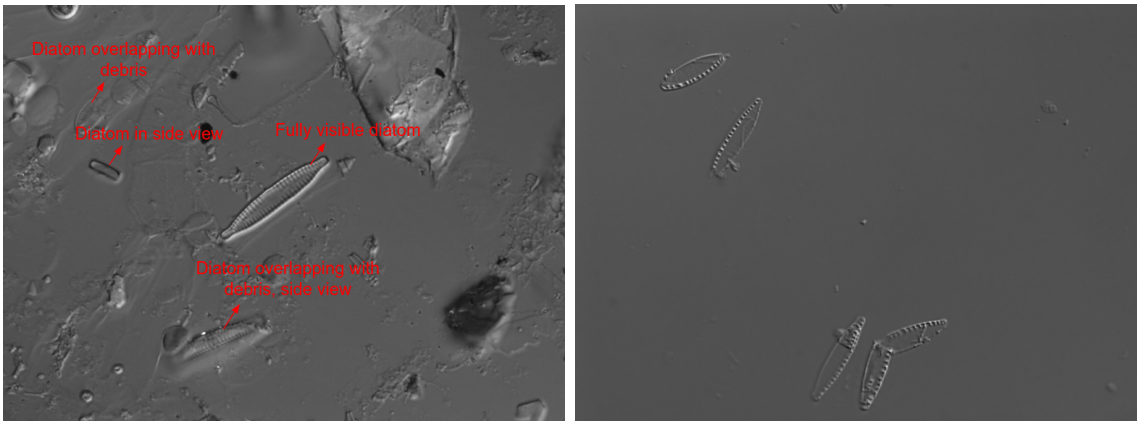


Figure 3.1: The image on the left is from the CLEURIE dataset, which represents a natural sample. In contrast, the image on the right is from the NPAL dataset and depicts a monospecific sample composed solely of a single diatom species, NPAL. (Left) Depending on the focal length of the microscope used to acquire the images, some diatoms are hardly discernible due to the lack of contrast between the background and the diatoms. (Left and Right) Diatoms are occluded by nearby diatoms or debris, making it challenging to extract the fully visible diatoms automatically.

An additional challenge arises when using supervised deep learning networks, which is, training requires many labelled examples. To achieve good performance from deep learning networks, one needs several hundred to thousand ground truth examples, which is usually obtained manually. In addition to being a time-consuming and laborious process, human intervention is also a source of error and bias [156]. Moreover, obtaining such huge amounts of datasets can be impractical. In this work, we use synthetic datasets, which are an attractive alternative to circumvent the need for annotating thousands of images. Thousands of these synthetic images can be generated within a few hours. The synthetic datasets were generated for both bounding box detection and semantic segmentation. In the first part of the chapter, we discuss the influence of these synthetic datasets on the bounding box detector's performance. We analysed the impact of the different parameters used to generate the datasets through a sensitivity analysis. For the evaluation on real datasets, we used the CLEURIE and NPAL datasets, as described in Section 1.2.1. This work was published in the journal *Engineering Applications of Artificial Intelligence* [180] and was done in collaboration with Pierre Faure-Giovagnoli as part of his Master's thesis [51]. The dataset and code is available in https://doi.org/10.12763/UADENQ. In the second part, we analysed the performance of semantic segmentation networks for diatom segmentation. Finally, we applied bounding box detection and semantic segmentation to perform instance segmentation on the NPAL dataset. This was further used to extract the diatom morphological parameters and to study the impact of the micro-pollutant Glyphosate on their morphology. This work was done in

collaboration with Sarah Chéron as part of her PhD thesis, and with Clara Martinez and Huyên Phong Hamon as part of their master thesis. This work will be submitted for a journal publication (Chéron et al., in preparation cf Section. 1.4).

3.2 Related Works

The first part of this section reviews the various methods employed for microorganism detection. The second part summarizes the various approaches to deal with scarce labelled data while training supervised deep learning networks.

3.2.1 Detection methods for aquatic organisms

This section reviews the detection methods that are employed in aquatic microorganism and plankton detection.

Bounding box detection

[157] compares the performance of a classical method: Viola-Jones detector [184] and YOLO [148], a deep learning bounding box detector. Axis-aligned bounding boxes still do not solve the problem of overlapping detection in the presence of closely located objects. This is a common problem in overhead imagery detection, and one way to ameliorate this is by using rotated bounding boxes. [107, 208] propose rotated bounding box detection methods using deep CNNs for ship detection in remote sensing. Apart from the bounding box coordinates learnt by the CNN, it learns an additional parameter to determine the angle of rotation of the detection box. In this work, we evaluated both axis-aligned and rotated bounding boxes for diatom detection.

Semantic and Instance Segmentation

Semantic segmentation is ubiquitous in microorganism image analyses, and therefore, several approaches on segmentation have been attempted. Classical methods for semantic segmentation include a combination of thresholding, gradient based, deformable models and feature based methods to perform the segmentation. [15] uses level sets, a numeral tool that is used to analyze shapes, to segment green microalgae. [59] uses an active contour model to generate multiple proposals of object contours, and uses machine learning to score each of these contours to perform phytoplankton segmentation. [91] introduces a toolbox to perform diatom segmentation. It includes Otsu's thresholding and Canny edge detection, combined with various shape optimization techniques. [183] uses phase congruency to detect circular

objects, combined with machine learning based classifiers to detect phytoplankton. While these methods achieved good performance, they are semi-automatic, requiring manual tuning of parameters, and not easily generalizable to organisms other than the ones that they are designed for, or different type of microscopy images. On the other hand, deep learning methods automatically learn the necessary features for detection, do not require extensive parameter tuning, and can easily be extended to different types of datasets.

One of the early attempts at using deep learning segmentation for diatoms is by [170], who uses DeepLab [26] to perform segmentation of *Chaetoceros*. [90] applies diatom segmentation on gigaslides. They analyse different tiling methods to break down the gigaslide to smaller images, suitable for training and obtaining predictions from deep learning models. A drawback of using semantic segmentation is that, if multiple individuals lie very close or overlap with each other, separating out the segmentation masks of each individual is very challenging. This can be overcome by instance segmentation methods, that combine the benefits of semantic segmentation and bounding box detection, and are finding widespread use in detection and object counting. [155] compares the performance between semantic segmentation and instance segmentation for diatoms and show that instance segmentation achieves a better performance. In this work, we used instance segmentation to extract diatom morphological parameters for morphometry analyses.

Morphometry Analysis

Traditional methods for morphometric analyses would rely on hand labelling the individual diatoms observed in the microscopy images and measuring the different parameter values of these diatoms using a general purpose image analysis software such as ImageJ [158]. This is a completely manual approach and is not very practical when the operation involves measuring the parameters of thousands of species. To facilitate the detection and parameter extraction for diatoms, SHERPA was introduced [91]. SHERPA allows for high volume examinations while reducing the amount of manual work required. This tool employs an image processing workflow that focuses on identifying and measuring object contours, handling all image segmentation steps from object identification to the extraction of morphological descriptors such as length, width, area, and perimeter. The microscopic images are provided to SHERPA which detects the contour of each diatom. To validate the contour, it is compared to a set of models representing desired shapes. The resulting selected outcomes are exported with a set of descriptors to enable further morphometric analysis. The detection pipeline of SHERPA is based on hand-crafted segmentation method, and Otsu thresholding. This brings us back to the earlier problem of using hand-crafted methods, namely, the requirement of manual intervention in tuning the parameters for accurate segmentation, lack of robustness and are

not suitable for detecting low contrast objects. Moreover, SHERPA has been specifically designed to work on bright field images, in contrast to the DIC images used in the thesis. Motivated by this, we leveraged instance segmentation using deep learning to extract the parameters from the masks.

3.2.2 Training Deep CNNs with scarse data

Several methods have been proposed to deal with the problem of data scarcity for training deep learning models. A widely used approach is transfer learning, where a model is trained on an available large dataset. The weights learnt by the model on this dataset is used to fine-tune to the smaller dataset at hand. This approach boosts the performance over models that have been trained only with small datasets [118,119,153]. Apart from transfer learning, a commonly adopted approach is to use data augmentation techniques. Data augmentation methods involve adding various transformations to the existing data to create new examples. These transformations can include various techniques such as cropping, flipping, rotating, zooming, changing the brightness or contrast, adding noise or distortions. By generating new data in this way, the model is exposed to a wider range of examples and can learn to generalize better, improving its accuracy and robustness. [174] addresses the problem of data scarcity in diatom classification by using augmentation techniques involving image morphing and image registration to generate new diatom data samples. Semi-supervised learning is a class of learning methods that leverages both labelled and unlabelled data to improve the performance of a learning model [197]. The addition of unlabelled data aids in capturing the underlying data distribution and extracting useful features, which in turn helps the model make more accurate predictions on the labelled data. Another line of work uses generative models such as GANs [64] to generate synthetic datasets [8, 188]. These methods generate fully synthetic data, where no part of the generated images contain real images. However, to generate high-quality images, these networks require thousands of images to train on, which is impractical for applications. In this thesis, we propose a method to generate partially synthetic data, (*i.e.,* where parts of the generated images contain real images). In the case of freshwater benthic diatoms, hundreds of digitized individual images are publicly available for various diatom species. In theory, they can also be retrieved in order to create thousands of synthetic images similar to real digital ones taken by a human expert. This offers a unique opportunity for the design of a relevant pre-labelled dataset for bounding boxes, as each individual is already boxed, allowing the training of automatic detection networks. Our method for synthetic data generation closely follows Copy-Paste augmentation as implemented for instance segmentation [46,60]. In Copy-Paste augmentation, the generated images use real images as background, then images of individual

objects of interest are pasted at random locations on the real images. However, while pasting, the global consistency of the final images is not considered. This means that an object could be placed at locations where they usually do not occur. This could be detrimental in the case of single-stage architectures such as YOLO [148], which predicts from full images and hence, takes into account the global context of the images while making predictions. By contrast, we generate a synthetic background and place the individual diatoms and debris images, while considering the global consistency of the final image. To generate realistic images, we consider different parameters while pasting individual objects, such as scaling, rotation, level of overlap of diatom and diatom-debris ratio.

3.3 Synthetic Dataset Generation

This section describes the procedure used to generate synthetic datasets for training the detection networks. To get the best model performance, the synthetic dataset should closely mimic the real images obtained from light microscopy.

Let us consider the following procedure to create a seamless image I of width W and height H with images from the individual Atlas and Debris dataset D:

1. Init I', empty image of dimensions (W, H).
   - Pick a subset $S_i \subset D$ composed of individual diatom and debris images.
   - For each image $T_{ij} \in S_i$ of width w_{ij} and height h_{ij},
     - Apply data augmentations: scaling, rotation, overlap and diatom-debris ratio. Refer to Sec. 3.4.1 for the different augmentations and the parameters used.
     - Find a random position P_{ij} for T_{ij} on I' such that it does not overlap any previously added diatom or debris images or the borders of I' by more than a certain threshold percentage.
2. Init I, empty image of dimensions (W, H). Let B be the set of pixels $p^{I'}$ of I' comprising the borders of T_{ij}. Set the value of each pixel p_i^I of I as a weighted arithmetic mean such that:

$$p_i^I = \frac{\sum_{p_j^{I'} \in B} val(p_j^{I'}) w(p_i^I, p_j^{I'})}{\sum_{p_j \in B} w(p_i^I, p_j^{I'})} \tag{3.1}$$

$$w(p, p') = e^{-||p'-p||_2} \tag{3.2}$$

where $val(p)$ is the gray value of pixel p. This step generates the gray background, similar to those obtained from light microscopy.

3. Finally, place T_{ij} at P_{ij} on I using the Poisson Editing method for seamless blending [141].

The pipeline for generating synthetic datasets for diatom detection is illustrated in Figure 3.2.

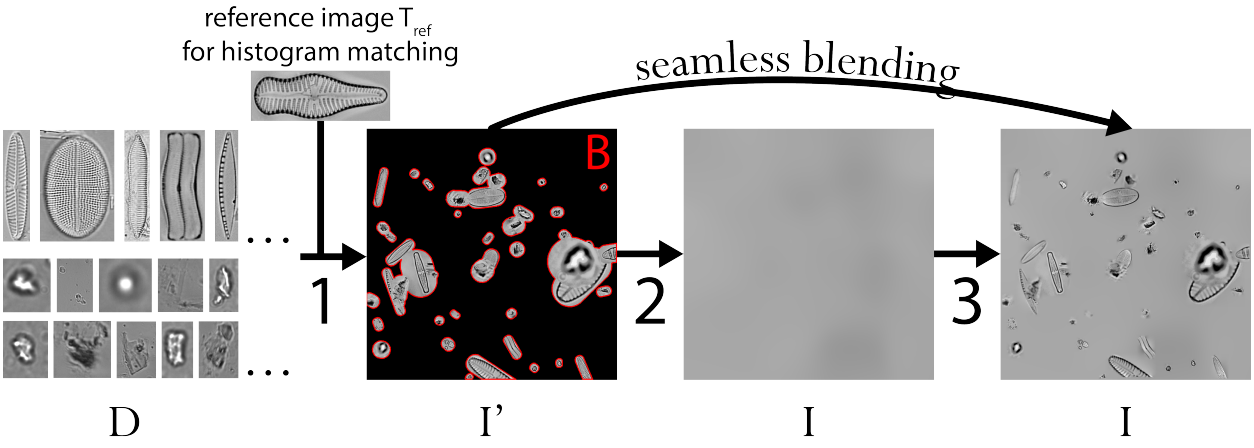


Figure 3.2: Illustration of the synthetic data generation process. The illustrated pipeline is used to generate data for both bounding box detection and semantic segmentation.

Since the individual diatom images are tightly cropped, the ground truth bounding coordinates can be obtained straightforwardly: the length and width of the bounding boxes correspond to the length and width of the individual diatom images. The box coordinates can be obtained based on the position P_{ij}, box dimensions and the rotation angles.

For semantic segmentation, it is assumed that the regions corresponding to diatoms are known. We obtain this by hand labelling the individual diatom images into regions of "Diatom" and "Non-diatom". The obtained binary masks are used to create the ground truth masks.

3.4 Experiments on bounding box detection

In this section, we present a sensitivity analysis, examining how variations in synthetic dataset parameters impact the automatic diatom detection method, focusing on bounding box detection.

3.4.1 Sensitivity Analysis of Parameters in Synthetic Dataset Generation

Data Augmentation and Baseline

Four augmentation techniques were used in order to generate the synthetic dataset: scaling, rotation, overlap between diatoms and debris, and diatom to debris ratio. To determine the sensitivity of each of the augmentation methods while evaluating using real microscopy images, and to choose the optimal parameter values, we performed a grid-search on the parameters without changing the neural network hyperparameters as summarized in Table 3.1. The definition and significance of each of these techniques and the parameter values are given below.

Dataset name	Parameters
D1	No rotation; Scaling factor=1; Diatom-debris ratio=[2,5]; Overlap=0;
D2	No rotation; Scaling factor=1; Diatom-debris ratio=[2,5]; Overlap=0.25;
D3	No rotation; Scaling factor=1; Diatom-debris ratio=[0.1,0.5]; Overlap=0;
D4	No rotation; Scaling factor=1; Diatom-debris ratio=[0.1,0.5]; Overlap=0.25;
D5	No rotation; Scaling factor=exponential; Diatom-debris ratio=[2,5]; Overlap=0;
D6	No rotation; Scaling factor=exponential; Diatom-debris ratio=[2,5]; Overlap=0.25;
D7	No rotation; Scaling factor=exponential; Diatom-debris ratio=[0.1,0.5]; Overlap=0;
D8	No rotation; Scaling factor=exponential; Diatom-debris ratio=[0.1,0.5]; Overlap=0.25;
D9	With rotation; Scaling factor=1; Diatom-debris ratio=[2,5]; Overlap=0;
D10	With rotation; Scaling factor=1; Diatom-debris ratio=[2,5]; Overlap=0.25;
D11	With rotation; Scaling factor=1; Diatom-debris ratio=[0.1,0.5]; Overlap=0;
D12	With rotation; Scaling factor=1; Diatom-debris ratio=[0.1,0.5]; Overlap=0.25;
D13	With rotation; Scaling factor=exponential; Diatom-debris ratio=[2,5]; Overlap=0;
D14	With rotation; Scaling factor=exponential; Diatom-debris ratio=[2,5]; Overlap=0.25;
D15	With rotation; Scaling factor=exponential; Diatom-debris ratio=[0.1,0.5]; Overlap=0;
D16	With rotation; Scaling factor=exponential; Diatom-debris ratio=[0.1,0.5]; Overlap=0.25;

Table 3.1: Augmentation methods and the parameters used to generate synthetic datasets

1. **Scaling**: Scaling refers to resizing the images of individual diatoms and debris while creating the synthetic dataset. This was carried out to represent the real microscopy images where diatoms belonging to a species can occur at varying scales. For an accurate scaling value, one could consider the actual size range in which the diatoms occur

naturally. However, for our analysis, we randomly chose a scaling value from an exponential distribution to consider a wide range of scales for each diatom species. For the grid-search, we considered situations with and without scaling to generate the data.

2. **Rotation**: The individual diatom images were randomly rotated at any angle between 0 and 360°. Our grid-search considered two conditions: with and without rotating the images.

3. **Overlap**: For each individual diatom on the synthetic image, this refers to the level of overlap allowed between other diatoms or debris. We varied the level of overlap between the objects from 0 to 25%. The maximum threshold value was chosen as 25% to avoid hidden objects that would bias training and evaluation. For our parameter-tuning, we considered two cases: no overlap between the objects and overlap between objects.

4. **Diatom-debris ratio**: This refers to the ratio between the number of diatoms and the number of debris objects while generating the images. We considered the following range of values:(1) diatom-debris ratio between 2 and 5; (2) diatom-debris ratio between 0.1 and 0.5. For the above two cases, the number of diatoms was between 6 and 10, and we varied the number of debris objects to achieve the desired ratio.

To study the impact of the synthetic dataset while training on the real datasets, we performed the following experiments:

1. **Synthetic -** The network was trained and evaluated on the synthetic images

2. **Zero-shot learning -** The network was trained on the synthetic dataset and evaluated on the real microscopy images

3. **Fine-tuning -** The model trained on the synthetic dataset is fine-tuned using the real dataset and evaluated on the real dataset.

4. **Training from scratch -** The network was trained using the real images and evaluated on the real images

Figure 3.3 illustrates the pipeline for training and evaluation. The experiments for axis-aligned and rotated bounding box detection were performed using YOLOv5 [2, 81].

Datasets

For training and evaluation on real images, CLEURIE and NPAL datasets were used, consisting of 120 and 175 images, respectively. 80% of the labelled images were used for training and 20% for evaluation.

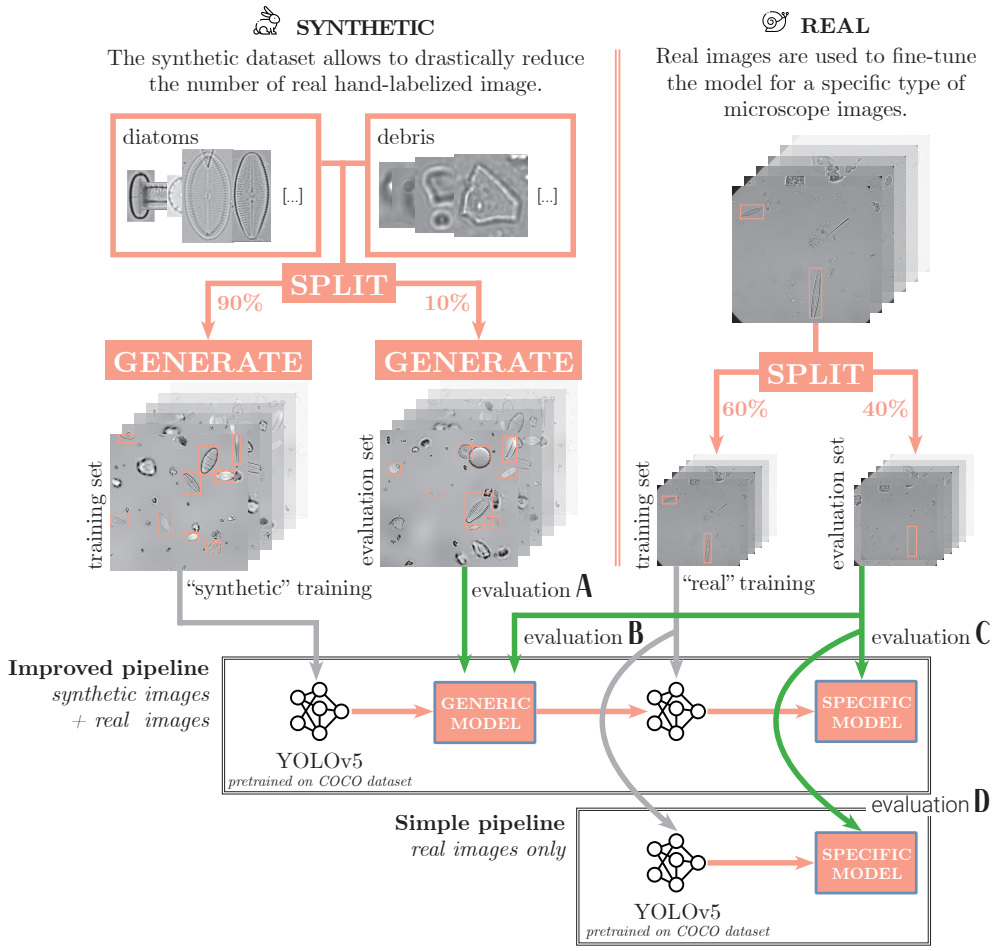


Figure 3.3: Pipelines for evaluating networks. First, individual diatoms and debris images are used to generate synthetic datasets for training the detection networks. There are four evaluation pipelines: Evaluation A – the network is trained and evaluated on synthetic datasets. Evaluation B – the network is trained on synthetic images and evaluated on real images (zero-shot detection). Evaluation C – the network is trained on a synthetic dataset, fine-tuned to real images, and evaluated on real images. Evaluation D – the network is trained and evaluated on real datasets

Evaluation Metrics

Detection is concluded when the intersection over union (IoU), which measures the extent of overlap between the predicted box and the ground truth box, is above a certain threshold. For a fixed threshold of IoU, precision gives the percentage of detections that are correct.

$$\text{Precision} = \frac{\text{True Positive}}{\text{True Positive} + \text{False Positive}} \tag{3.3}$$

For a fixed IoU threshold, recall gives the number of positive detections out of all the positives in the ground truth.

$$\text{Recall} = \frac{\text{True Positive}}{\text{True Positive} + \text{False Negative}} \tag{3.4}$$

In addition to these metrics, some Object Detection challenges (VOC, COCO, Open Images) define their own metrics that have become standard evaluation indicators. In this work, we also calculated mean Average Precision for Intersection over Union (IoU) at 0.5 (mAP@0.5) and AP for IoU from 0.50 to 0.95 with a step size of 0.05 (mAP@0.5:0.95) as defined in [34]. We used the same metrics for evaluating the axis-aligned and rotated bounding boxes.

Implementation Details

We used Adam optimizer [89] and the learning rate was 0.0002. All the experiments were performed on image dimensions of 512×512. The batch size for training YOLOv5 (axis-aligned and rotated) was 16. The training was performed for 100 epochs. For every category of synthetic dataset (D1-D16 in Table. 3.1), we generated 4000 images. While training the networks using synthetic and real datasets, we used 80% of the images for training, 10% for validation and 10% for testing. We trained our networks on GeForce GTX 2080 with 8 Gb RAM and GeForce GTX 1080 with 12 Gb RAM. The total training time for sensitivity analysis was 24 days.

3.4.2 Results

Synthetic Datasets Generation

Figure 3.4 shows examples of synthetic images generated for each augmentation method and parameters used in the sensitivity analysis. Refer to Table 3.1 for a recall of these dataset types.

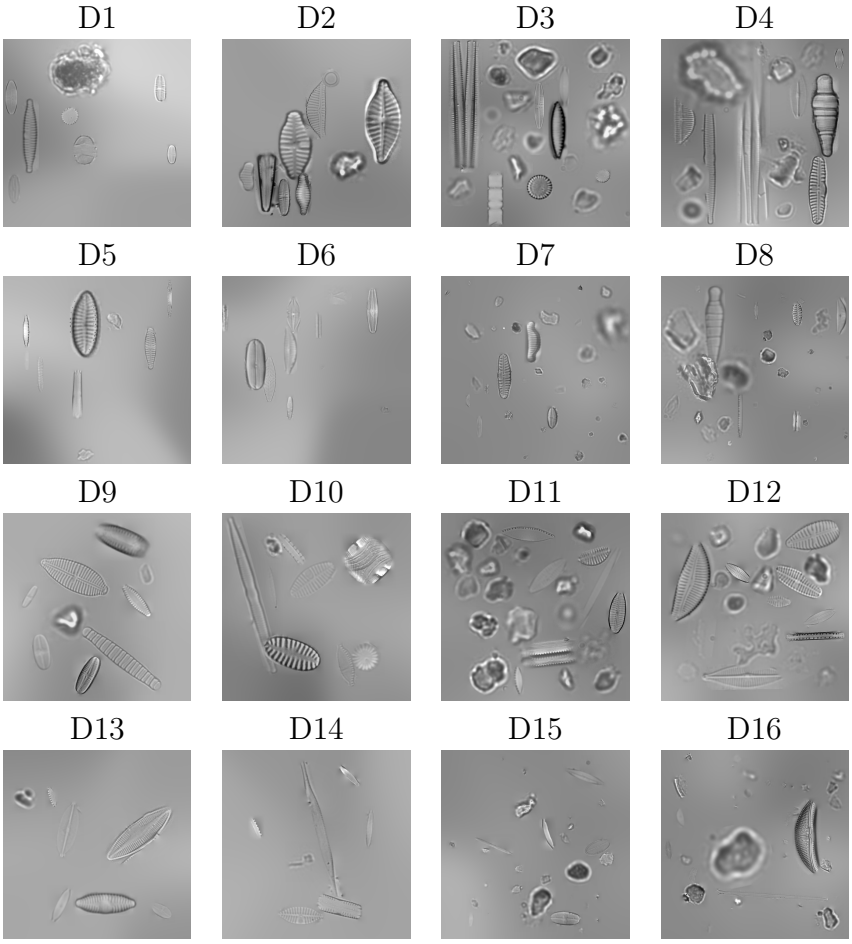


Figure 3.4: Examples of synthetic datasets generated for each of the augmentation parameters (D1-D16).

Axis-aligned bounding box detection

Figure 3.5 shows some examples of the axis-aligned bounding box detections from YOLOv5.

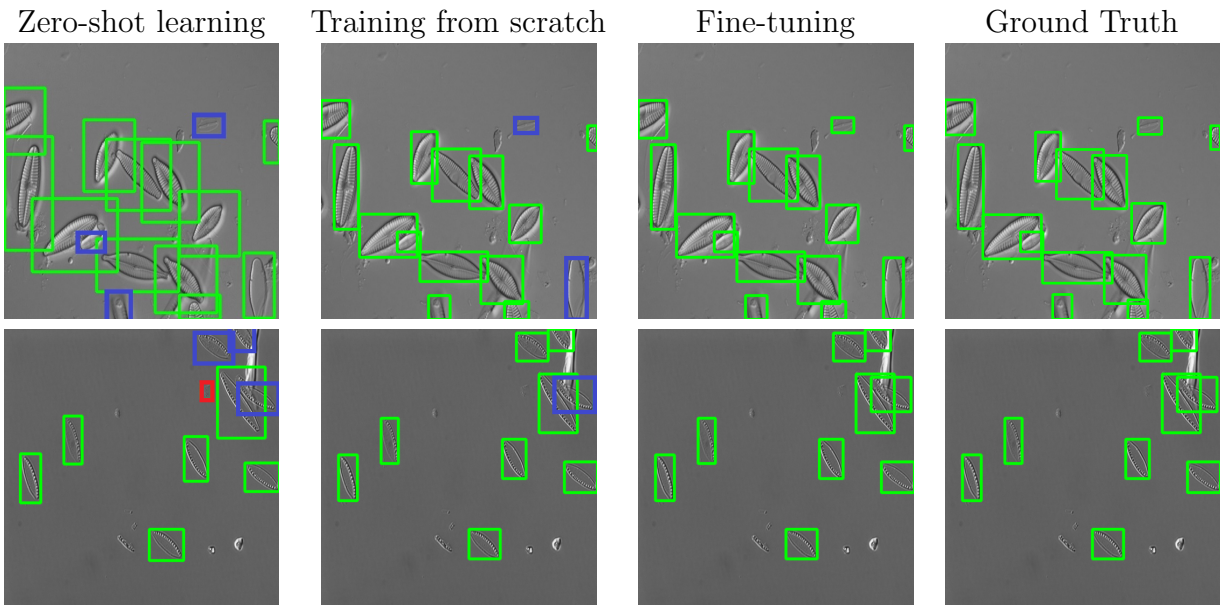


Figure 3.5: Axis-aligned bounding box detection obtained from YOLOv5 using CLEURIE (top row) and NPAL (bottom row) real images. Green boxes represent the correct detection; blue indicates false negatives and red boxes indicate false positives.

Table 3.2: Quantitative evaluation of axis-aligned bounding box detection. Higher values indicate better performance. M1-Precision; M2-Recall; M3-mAP@.5; M4-mAP@.5:.95.

Datasets	Synthetic				Zero-shot learning								Fine-tuning							
					CLEURIE				NPAL				CLEURIE				NPAL			
	M1	M2	M3	M4	M1	M2	M3	M4	M1	M2	M3	M4	M1	M2	M3	M4	M1	M2	M3	M4
D1	**1.000**	**1.000**	0.995	**0.990**	0.422	0.461	0.406	0.174	0.848	0.549	0.694	0.413	0.930	0.917	0.974	0.797	0.816	**0.958**	0.918	0.756
D2	0.998	0.998	0.995	0.981	0.632	0.453	0.501	0.224	0.533	0.690	0.603	0.405	0.925	0.923	0.974	0.804	0.842	0.901	0.909	0.745
D3	0.999	**1.000**	0.995	0.973	0.573	0.458	0.512	0.222	0.543	0.620	0.567	0.310	0.939	0.920	0.975	0.802	**0.936**	0.829	**0.933**	**0.786**
D4	0.998	0.999	0.995	0.967	0.605	0.605	0.614	0.254	0.639	0.775	0.683	0.257	0.921	0.934	0.976	0.802	0.837	0.944	0.924	0.775
D5	0.993	0.996	0.995	0.971	0.543	0.535	0.451	0.107	**0.861**	0.795	0.788	**0.308**	0.928	**0.963**	0.972	0.791	0.747	0.915	0.812	0.705
D6	0.995	0.991	0.995	0.968	0.579	0.404	0.444	0.195	0.475	0.423	0.464	0.275	0.933	0.923	0.979	0.791	0.805	0.928	0.903	0.747
D7	0.995	0.992	0.995	0.967	0.512	0.596	0.568	0.308	0.800	0.450	0.547	0.373	0.919	0.946	0.978	0.797	0.812	0.915	0.920	0.779
D8	0.995	0.991	0.995	0.965	0.802	0.513	0.645	0.353	0.766	0.507	0.531	0.284	0.907	0.948	0.971	0.786	0.840	0.887	0.892	0.770
D9	1.000	0.999	0.995	0.955	0.666	0.610	0.652	**0.364**	0.646	0.436	0.461	0.265	0.960	0.900	0.977	0.798	0.793	0.971	0.879	0.757
D10	0.998	0.991	0.995	0.915	0.791	0.642	0.675	0.173	0.805	0.761	0.741	0.253	0.910	0.954	0.977	0.791	0.767	0.972	0.850	0.723
D11	0.999	1.000	0.995	0.932	0.783	0.613	0.621	0.121	0.753	0.690	0.669	0.210	0.916	0.940	0.977	0.787	0.775	0.972	0.871	0.712
D12	0.999	**1.000**	**0.999**	0.946	0.695	0.642	0.670	0.190	0.762	0.859	0.804	0.175	0.910	0.954	0.974	0.784	0.802	0.915	0.899	0.744
D13	0.988	0.990	0.995	0.954	0.681	0.484	0.546	0.194	0.844	0.789	0.861	0.199	0.931	0.928	0.977	0.788	0.841	0.901	0.890	0.728
D14	0.982	0.987	0.994	0.931	**0.800**	**0.700**	**0.776**	0.300	0.710	0.901	0.763	0.235	**0.967**	0.911	**0.981**	**0.806**	0.805	0.930	0.886	0.749
D15	0.982	0.984	0.996	0.943	0.651	0.648	0.630	0.157	0.849	**0.958**	**0.913**	0.244	0.917	0.948	0.975	0.771	0.779	0.943	0.873	0.753
D16	0.980	0.974	0.993	0.922	0.766	0.685	0.750	0.211	0.766	0.924	0.858	0.275	0.933	0.920	0.979	0.789	0.800	**0.958**	0.878	0.738

Table 3.3: Quantitative evaluation of axis-aligned bounding box detection when the network is trained using only the real dataset.

Dataset	Precision	Recall	mAP@0.5	mAP@0.5:0.95
CLEURIE	0.840	0.894	0.864	0.611
NPAL	0.744	0.864	0.847	0.594

Tables 3.2 and 3.3 show the quantitative metrics for axis-aligned bounding box detection using YOLOv5 [81]. The following observations could be made based on the results:

Scaling: When evaluated on synthetic dataset, the precision, and recall (mIoU=0.50 and mIoU from 0.50 to 0.95) was higher when the images were not scaled (D1-D4; D9-D12 in Table. 3.2). This is because when there is no scaling, the diatoms are of uniform size, which leads to network overfitting. When evaluated on the real microscopy images, the CLEURIE dataset contained diatoms of varying size due to the presence of different species, and hence the presence of scaling in the synthetic dataset leads to better metric values on both zero-shot learning and fine-tuning. NPAL images contained diatoms of the same species, and all diatoms were of rather uniform size. Hence, the network generalized better when the synthetic dataset had no scaling.

Rotation: When the dataset contained no rotated diatoms (D1-D8), the $\text{mIoU}_{0.5:0.95}$ on the synthetic dataset was superior, but it did not generalize well to the CLEURIE and NPAL datasets. This is because networks detecting axis-aligned bounding boxes are designed to fit non-rotated objects better, hence the network generalizes well. The network draws tighter and more accurate bounding boxes around these objects, resulting in better mIoU values. However, real datasets contain diatoms present in random orientations. Thus, training datasets D9-D16 generalized better to CLEURIE and NPAL datasets compared with D1-D8.

Overlap: From the results, having overlaps had no significant impact on the precision and recall scores (mIoU=0.50 and mIoU from 0.50 to 0.95) of the network for synthetic and real datasets. However, the best metric values were obtained when a maximum overlap of 25% was allowed. This means that although overlap helps improve network performance, the network was able to generalize well even when the training datasets do not have overlap of diatoms.

Diatom-debris ratio: The network had better metrics when the diatom-debris ratio was in the range [2,5] for both the synthetic and real datasets. For the synthetic datasets, this could be explained by the reduced complexity of the images for the network to detect, since the debris objects were widely distributed. For both the CLEURIE and NPAL datasets, the diatom-debris ratio was greater than 1. Hence, when the synthetic dataset was characterized by a higher diatom-debris ratio, the network generalized better for these datasets.

Table 3.3 shows the results when the network was trained using only the real dataset.

Compared with the performance when the network was trained only on real data, there was an improvement of 15% in precision (mIoU=0.50) and 8% in recall (mIoU=0.50) for the CLEURIE dataset and an improvement of 25% in precision (mIoU=0.50) and 11% in recall (mIoU=0.50) for the NPAL dataset when the network was trained on synthetic data and fine-tuned to real data.

Rotated Bounding Box Detection

Figure 3.6 shows some examples of the rotated bounding box detection from YOLOv5. Tables 3.4 and 3.5 show the quantitative metrics for rotated bounding box detection. Since no rotation was included in D1-D8, we considered datasets D9-D16.

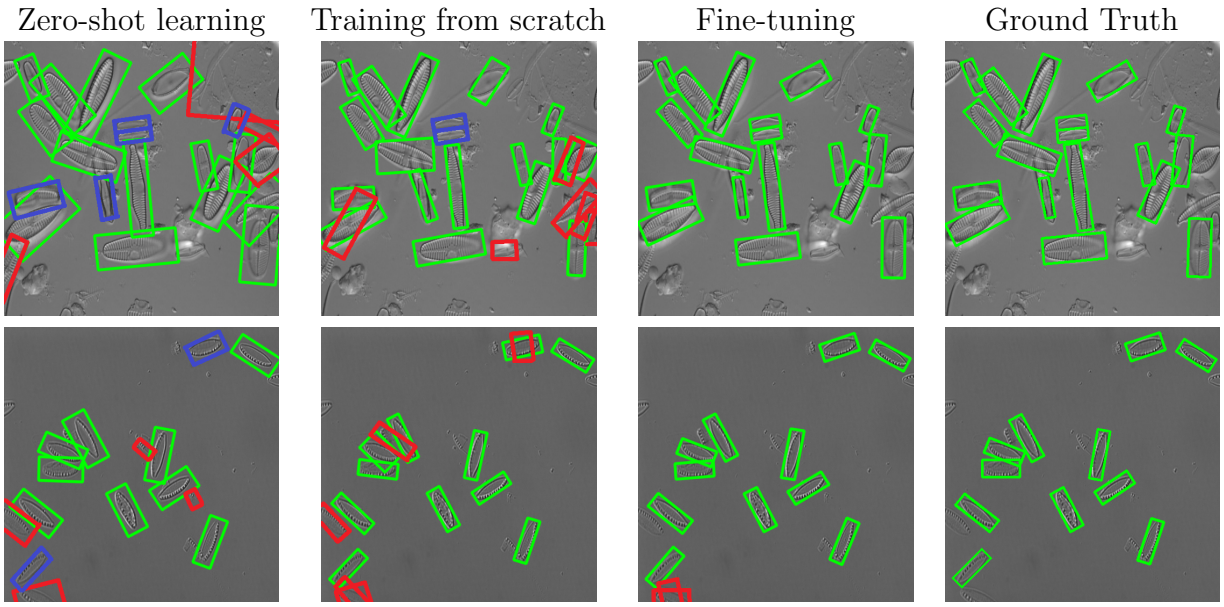


Figure 3.6: Rotated bounding box detection from YOLOv5 using CLEURIE (top row) and NPAL (bottom row) real images. Green boxes indicate correct detection; blue indicates false negatives, and red boxes indicate false positives.

Table 3.4: Quantitative evaluation of rotated bounding box detection. Higher values indicate better performance.
M1-Precision; M2-Recall; M3-mAP@.5; M4-mAP@.5:.95.

Datasets	Synthetic				Zero-shot learning								Fine-tuning							
					CLEURIE				NPAL				CLEURIE				NPAL			
	M1	M2	M3	M4	M1	M2	M3	M4	M1	M2	M3	M4	M1	M2	M3	M4	M1	M2	M3	M4
D9	**0.983**	0.992	**0.991**	**0.811**	0.302	0.320	0.172	0.037	0.634	0.706	0.618	0.198	0.868	0.958	**0.928**	0.679	0.862	0.955	0.939	0.724
D10	0.975	**0.987**	0.988	0.793	0.536	0.414	0.346	0.027	**0.813**	**0.806**	**0.804**	**0.234**	0.826	0.819	0.865	0.615	0.880	0.991	0.940	0.730
D11	0.968	0.958	0.985	0.750	0.520	0.475	0.372	0.027	0.755	0.838	0.781	0.176	0.868	**0.957**	0.925	**0.699**	0.879	0.982	0.942	0.737
D12	0.972	0.993	0.981	0.781	0.387	0.260	0.162	0.028	0.731	0.675	0.656	0.176	**0.944**	0.823	0.890	0.662	**0.909**	0.991	**0.968**	**0.769**
D13	0.957	0.949	0.969	0.722	0.349	0.199	0.122	0.062	0.677	0.644	0.605	0.154	0.910	0.839	0.857	0.603	0.872	0.982	0.955	0.733
D14	0.953	0.941	0.958	0.721	0.393	0.398	0.268	0.049	0.640	0.665	0.605	0.154	0.927	0.823	0.857	0.626	0.880	**1.000**	0.937	0.715
D15	0.936	0.889	0.900	0.709	0.466	0.434	0.317	0.072	0.699	0.709	0.670	0.191	0.892	0.806	0.817	0.589	0.879	0.982	0.928	0.708
D16	0.936	0.876	0.903	0.680	**0.541**	**0.531**	**0.448**	**0.091**	0.736	0.803	0.748	0.221	0.846	0.887	0.875	0.647	0.908	0.982	0.952	0.743

Table 3.5: Quantitative evaluation of rotated bounding box detection when the network is trained using only the real dataset.

Dataset	Precision	Recall	mAP@0.5	mAP@0.5:0.95
CLEURIE	0.804	0.783	0.834	0.528
NPAL	0.843	0.811	0.871	0.642

Scaling: The evaluation metrics follow a similar trend to the axis-aligned box evaluation. Evaluation on the synthetic dataset achieved the best performance when there was no scaling, since the network overfits to the uniform diatom sizes. Scaling helps the network generalize better to the CLEURIE dataset due to the presence of different taxa of diatoms with varying scales. The NPAL dataset consists of diatoms of the same species with uniform sizes, hence the network generalizes best when trained with no scaling.

Overlap: When evaluated on the synthetic dataset, the network performed best when there was no overlap between the diatoms and debris. This is because, when there are no overlaps, there is no ambiguity due to occlusions for the network while learning the rotated bounding box parameters. For the CLEURIE and NPAL datasets, overlap helps the network generalize better.

Diatom-debris ratio: The network performed better when the diatom-debris ratio was in the range [2,5] for the synthetic and real datasets. This is because fewer debris objects means a sparser distribution compared with the diatoms, making them easier to discard.

Table 3.5 shows the results when the network was trained using only the real dataset. Compared with the performance when the network was trained only on real data, there was an improvement of 17% in precision and 22% in recall for the CLEURIE dataset and an improvement of 8% in precision and 23% in recall for the NPAL dataset, for an mIoU threshold of 0.5 when the network was trained on synthetic data and fine-tuned to real data.

3.5 Experiments on automatic diatom segmentation

In this section, we investigate the use of deep learning for automatic segmentation of the diatoms. We use instance segmentation for extracting the individual NPAL diatoms. As a recall, instance segmentation involves simultaneously performing segmentation and rotated bounding box detection, which gives the masks of the individual diatom extracted. Using the masks, one can extract the various morphological parameters for the morphometric analyses.

Our study focuses on investigating the impact of Glyphosate, a herbicide, on diatom morphology. The hypothesis behind the experiment is that the exposure of the diatoms to these chemicals could induce an inhibition of the growth of diatoms, and cause severe frustule

Table 3.6: Table summarizing the number of images acquired for the replicates of each chemical treatment.

Chemical	Concentration	Replicate no.	Number of images
Glyphosate	IC50	1	194
		2	102
		3	156
	Control	1	175
		2	96
		3	134

deformations compared to unexposed diatoms. To test this, cultures of NPAL were exposed to Glyphosate, and their morphological parameters were analysed to test for variations when compared to the non-exposed NPAL cultures.

3.5.1 Materials and Methods

Dataset Acquisition

The dataset under consideration consists of monospecific, *i.e.,* organisms of only one species of diatoms: *Nitzschia palea* (NPAL). NPAL is very well studied, easily cultivable and maintainable in the laboratory. The organisms were exposed to glyphosate for 28 days. The tested concentration corresponded to the IC50 value of the contaminants, as well as to the environmental concentrations. 3 replicates were prepared for the given concentration.

The number of images acquired for the three replicates in each chemical treatment is provided in Table. 3.6.

Dataset Labelling

For training the segmentation network, a total of 857 images from the Glyphosate treatment (IC50 and control) were hand labelled using Labelbox. For the morphometric analyses, it is important to ensure that the masks belong to individual fully visible diatoms, as otherwise, the presence of overlapping or partial masks would lead to incorrect parameter estimates. To discriminate these types of diatoms from the rest, the labels were divided into four main categories: "Diatom", "Overlap", "Piece of Diatom" and "Other". Here, the "Diatom" category belongs to the fully visible individual diatoms, such that they do not touch or overlap with any neighbouring objects. The diatoms that overlap with other diatoms or debris were labelled as "Overlap", broken diatoms and those that were present on the image edges and thus, were not fully visible were labelled as "Piece of Diatom". Remaining portions were labelled "Other".

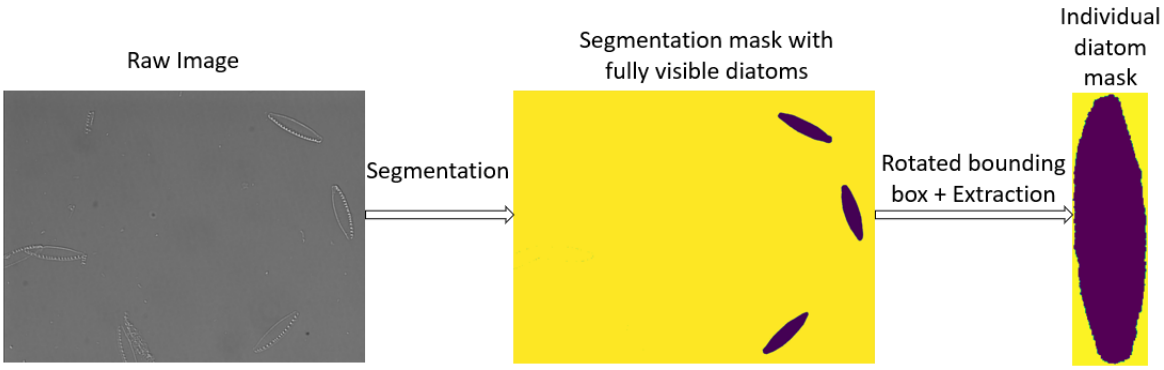


Figure 3.7: Figure illustrating the pipeline for individual diatom mask extraction.

For the rotated bounding box detection, the dataset described in 3.4.1 was used. The network was first trained on the synthetic dataset D12, and fine-tuned to the real dataset.

Diatom Parameter Extraction and Analysis

To study the influence of the micropollutants on NPAL, the morphological parameters of the individual diatoms must be extracted. Instance segmentation was used to extract the individual diatoms and their masks. To perform instance segmentation, a semantic segmentation network and rotated bounding box detection were used. The output from the segmentation network is a binary mask, representing the diatom and the non-diatom regions. The individual diatom masks are extracted using rotated bounding boxes. Figure. 3.7 illustrates the pipeline for extracting the individual diatom masks from a microscopy image.

Different parameters related to the geometry and texture of the diatoms were calculated. From the individual diatom masks, the contour of the diatom is used to calculate various morphological parameters to describe the size and shape. The parameters extracted were length, width, area, perimeter, ellipticity, eccentricity, roundness, convexity, concavity, rectangularity, and the Hu moments [75].

To extract the texture descriptors, Gray-Level Co-occurrence Matrix (GLCM) and Local Binary Patterns (LBP) were used. GLCM is a matrix that describes the distribution of co-occurring pixel values in a digital image. It is typically calculated by first defining a direction (such as vertical, horizontal, or diagonal) and a distance between pixels (also known as the offset). Then, for each pixel in the image, the value of the neighbouring pixel that is located at the specified offset and direction is recorded. This process is repeated for all pixels in the image, resulting in a matrix that represents the frequency of co-occurring pixel values for a given offset and direction. From the GLCM, the contrast, dissimilarity, homogeneity, correlation, and energy were calculated. Local Binary Pattern (LBP) compares the intensity

value of each pixel in an image with its neighbouring pixels. Specifically, LBP defines a binary code for each pixel in an image based on the comparison of its intensity value with the intensity values of its neighbours. The binary code is then used to represent the local texture of the image at that pixel location. The different LBP based statistical parameters extracted are the mean, minimum value, maximum value, first quartile, second quartile, third quartile, variance, standard deviation, skewness, kurtosis, and entropy. The definitions of the parameters are provided in Appendix. B.

It could sometimes happen that the detected masks obtained from the automatic detection system are incorrect. This can happen when there are false positives, where a non-diatom region is detected or the detected diatom is incomplete due to occlusions or broken. The incorrect detections were removed based on the eccentricity value estimated from the detected mask. Only the masks that have eccentricity > 0.99 were considered for the morphometric analyses.

Principal Component Analysis (PCA) was employed to represent the morphometric parameters of the diatoms and to assess the differences in morphology between diatoms exposed to glyphosate and those that were not exposed. The PCAs were conducted using the FactoMineR package [79] and the Factoextra package [86] in R Core Team (2022), version R 4.1.3.

Since the study couldn't adequately discern the ornamental features of frustules like striae and fibulae, these six shape descriptors were chosen for a comprehensive analysis of diatom valve morphology.

3.5.2 Experiments

Semantic Segmentation

Baseline. For semantic segmentation, we consider the following three experiments:

1. **Zero Shot Learning -** The network is trained on synthetic data and evaluated on real dataset.

2. **Fine-tuning -** Network trained on synthetic data is fine-tuned to the real dataset and evaluated on real dataset.

3. **Training from scratch -** The network is trained and evaluated on the real dataset.

The experiments were conducted on DeepLabV3 [27] with ResNet50 backbone [71].

Evaluation Metrics. For evaluation, we used the standard mean Intersection over Union scores (mIoU). The mIoU score is calculated as the mean intersection over union

(IoU) score across all objects or classes in the image. The IoU score measures the overlap between the predicted segmentation and the ground truth segmentation, and is calculated as the intersection of the two segmentations divided by the union of the two segmentations.

Implementation Details. We used Adam optimizer with a learning rate of 0.01. All the experiments were performed on image dimensions of 2040×1536 and a batch size of 2. The training was performed for 150 epochs. For the training using synthetic data, we use the data type D16, consisting of 4000 examples. We trained our networks on two GeForce GTX 3090 with 24 Gb RAM each. For the training, images in all the replicates of Glyphosate IC50 and replicate numbers 1 and 2 of Glyphosate Control were used for training, and the replicate number 3 of Glyphosate Control was used for evaluation.

Morphological Parameters Extraction

For analysing the performance of morphological parameters extracted using detection networks, we compare the parameters extracted using automatic detection with those extracted from the hand labelled masks. We compare the error between the two methods for five morphological parameters that are ecologically interpretable: length, width, area, perimeter, and roundness. Three other descriptors, namely "ellipticity," "convexity," and "concavity," were excluded from the study because they exhibited high correlations with at least one of the existing descriptors and did not contribute any supplementary information.

3.5.3 Results

Semantic Segmentation

The mIoU for the experiment on zero-shot learning was 56.32%, training from scratch was 89.37% and fine-tuning was 91.59%. One explanation for the low score on zero-shot learning could be because NPAL dataset consists of single diatom species, whereas the synthetic dataset consists of several types of species. The synthetic datasets are not very representative of the NPAL dataset. The difference in performance between the fine-tuning experiment and training from scratch is only marginal, with 2% improvement when using synthetic datasets. An explanation for this is the relatively large number of training images available for the NPAL dataset.

Examples of predictions obtained from the fine-tuning experiment are shown in Figure 3.8.

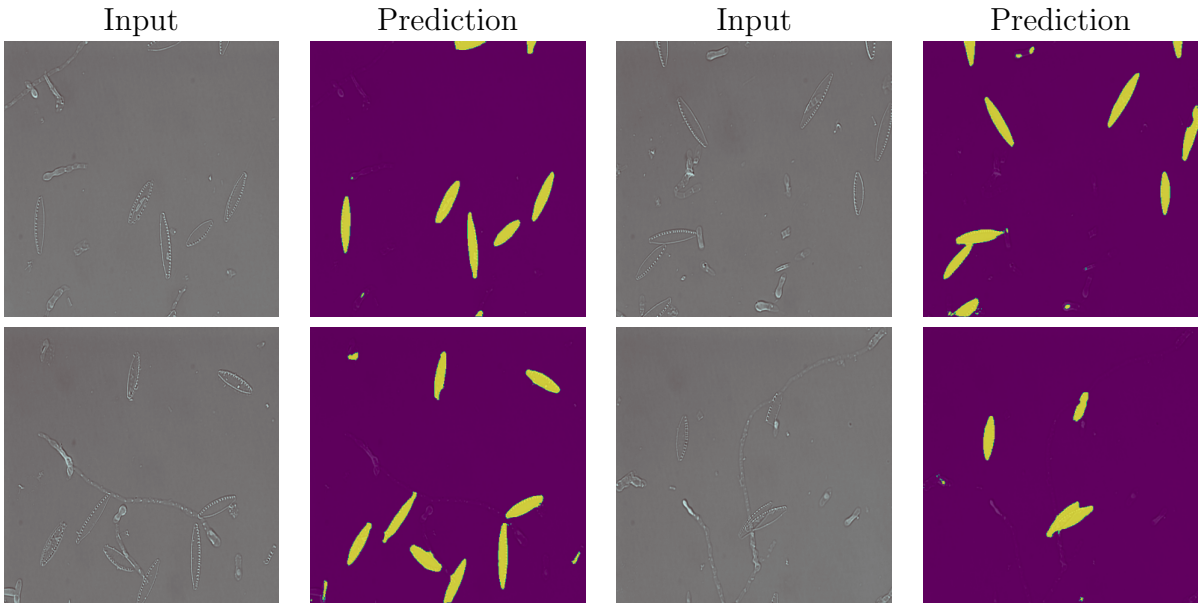


Figure 3.8: Predictions obtained for semantic segmentation of diatoms in the NPAL dataset. In the predictions, the yellow regions are predicted as diatoms and the purple regions are predicted as non-diatoms.

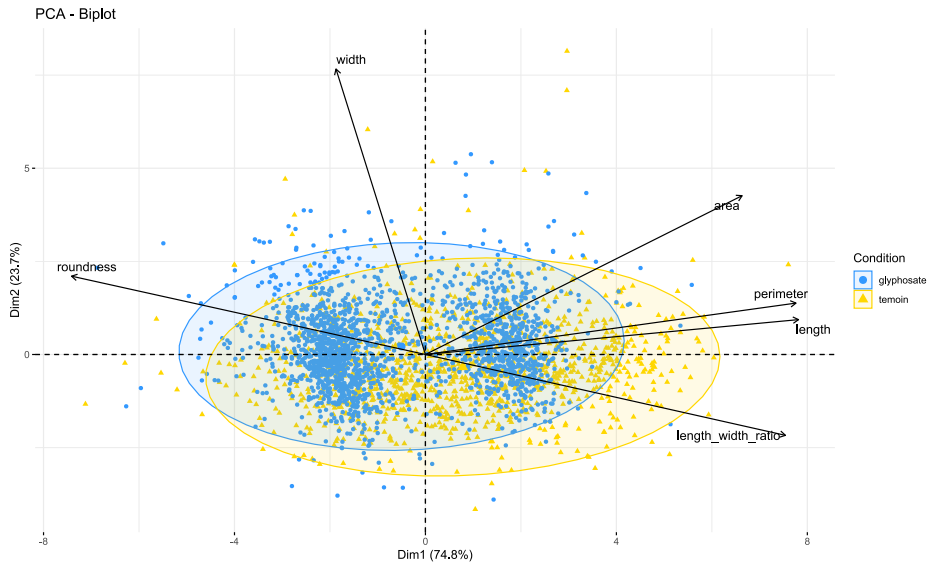


Figure 3.9: PCA performed on 6 morphometric descriptors measured automatically on NPAL exposed to IC50 values of glyphosate for 28 days. Each point corresponds to an individual valve. Yellow points represent control and blue points represent glyphosate condition. Confidence ellipses contain 95% of the individuals in a single condition.

PCA Analysis of the Morphometric Descriptors

In Figure 3.9, Principal Component Analysis (PCA) was conducted on six morphometric descriptors: length, width, length/width ratio, area, perimeter, and cell roundness of the diatom NPAL when exposed to the IC50 concentration of glyphosate. Since the fine details of frustules such as striae and fibulae were not clearly discernible for inclusion in the study, these six shape descriptors were selected for an in-depth analysis of diatom valve morphology.

The first two dimensions of the PCAs captured a substantial amount of data variability, accounting for 98.5% of the total variance. When visualizing the results with PCA ellipses, which encompass 95% of the individuals for each condition, it became evident that, on the whole, the morphological characteristics of control diatoms closely resembled those of diatoms exposed to glyphosate.

Individual masks and parameter extraction

Fig. 3.10 shows the bar plots of the error values obtained when comparing the parameters extracted using automatic detection with those extracted using manual labelling. Note that the values reported are in terms of number of pixels. Out of 407 individual diatoms manually labelled, 310 of them were detected by the network. From the results, the mean error values for all the parameters under comparison are below 6%. The low error rate suggests that automatic segmentation can serve as a reliable tool not only for detection, but also for extracting the various parameters needed for morphometric analyses.

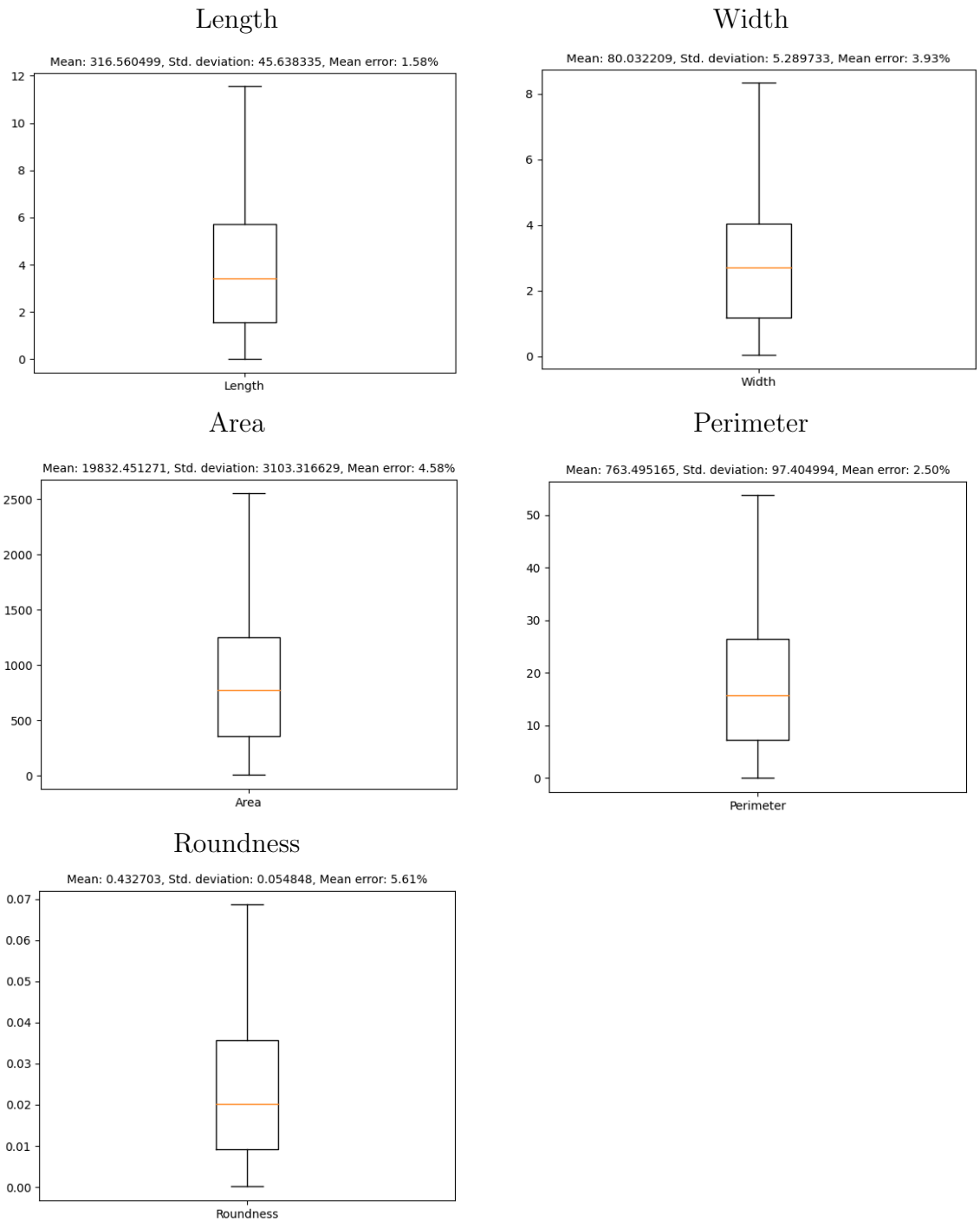


Figure 3.10: Box plot shoring the error between the morphological parameters estimated using automatic detection and hand labelled masks. The values in the y-axis of the plots show the error values. Note that the values are calculated based on the image pixels.

3.6 Conclusion

In this chapter, we presented the different detection methods using deep learning for automatic diatom detection, namely, bounding box detection and image segmentation. One of the main challenges faced while training deep learning based detection methods is the availability of high quality labelled datasets. Most of the time, the labelling needs to be done manually, making it a time-consuming and expensive process to obtain labelled datasets. To overcome this, we presented a method to generate synthetic datasets for the detection methods. Using this approach, thousands of synthetic microscopy images with ground truth labels can be obtained for training the different detection methods. We demonstrated the effectiveness of the different parameters used to generate the synthetic data through a sensitivity analysis on axis-aligned and rotated bounding box detection. Results show that when a limited number of labelled images are available for training, synthetic datasets can boost the performance of the deep learning methods.

Finally, we demonstrated an application of the segmentation and rotated bounding box detection to a use case on studying the impact of micropollutants on the diatom morphology. The detection methods can be used to extract the individual diatom masks and the different morphological and textural parameters, that can be used for further morphometric analyses. The results highlight that automatic detection tools can be a useful and effective tool to aid real life use cases involving detection or analysing specie morphology.

Chapter 4

Classification and Uncertainty Estimation

In the preceding chapter, we addressed the challenge of automating diatom detection. The subsequent phase within the pipeline involves identifying the taxonomy associated with the detected diatoms. This chapter deals with automatic taxonomic identification through the application of deep learning-based classification. The first part of the chapter addresses the problem of fine-grained classification. Given that diatoms exhibit significant inter-species similarity and intra-species variances in their morphology, we propose a method aimed at enhancing the classifier's performance, with a special focus on the specificities of the diatom dataset. The last part of the chapter addresses the challenge of improving a classifier's reliability by quantifying the uncertainty in their predictions. This is particularly important since deep learning classifiers tend to produce overconfident outputs when provided with Out-of-Distribution (OOD) data *i.e.,* data that lie beyond the scope of the training data distribution. An example of a situation where this will be required is when the diatom detector identifies false positives. We develop a method to quantify the uncertainty in the classifier's predictions, and further use this to identify Out-Of-Distribution (OOD) data. This chapter builds upon the work published in *ICVS 2021* [181] and *ICCV Workshop 2023* [179].

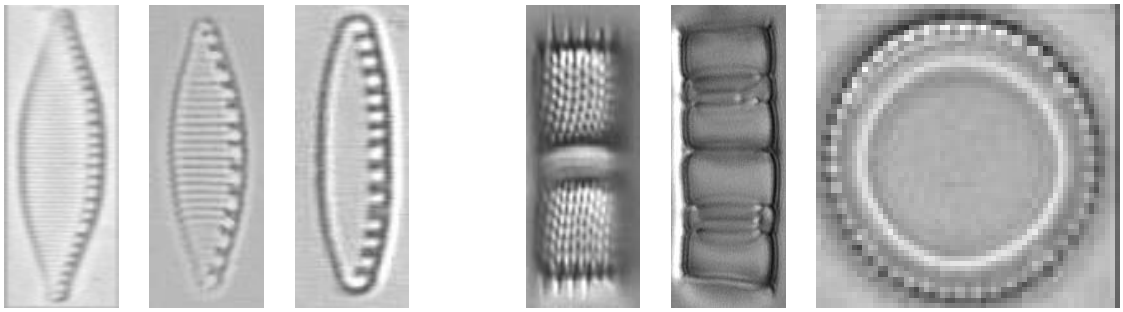
4.1 Tackling Inter-class Similarity and Intra-class Variance for Diatom Classification

One key requirement of diatom-based biomonitoring is to analyse the distribution of occurrence of the different diatom species. This often requires manual identification of the

organisms by human experts. Diatoms exhibit a wide range of morphological features, and they are grouped into taxa based on these features. With the huge number of diatom taxa identified, the differences in the morphological diversity among them are subtle. This often makes the classification task time-consuming, tedious and often subject to errors, even among taxonomists with high level of expertise. The challenge faced during manual classification stimulates the need to automate the classification. One of the early attempts at automating diatom identification is the ADIAC project [44]. In the ADIAC dataset, a set of 171 manually designed features were utilized for diatom classification. These features aim to capture various aspects of diatom characteristics, such as symmetry, shape, geometry, and texture. Over the years, the automatic classification has evolved from traditional hand-crafted methods [19, 204] to deep-learning-based ones [119, 137]. Instead of manually selecting the different features for identification, a CNN acts as an automatic feature extractor, learning the necessary features for identifying the different diatom species. While deep learning methods have shown improved performance over the handcrafted feature identification, there are still some challenges when classifying images with fine-grained visual features, as observed in diatoms [138].

Automated classification of diatoms is challenging, notably due to two aspects: inter-class similarity and intra-class variance. Inter-class similarity arises when objects from distinct classes exhibit visually similar appearances due to small variations in their morphological features, posing challenges in discerning the subtle differences. Intra-class variance refers to the situation where objects within the same class exhibit significant variations in their appearances, leading to distinct visual differences among them. In this case, the network fails to correctly identify all of them as members of the same class. This is prevalent in microscopic images due to the restrictions in view-points from which the images are acquired. Diatoms are typically imaged on permanent microscope slides: samples are treated chemically in order to remove organic materials and the diatom suspension is allowed to settle out, which restricts each organism to lie either on its side or top. Due to the imaging constraints, images acquired using these microscopy methods result in discrete view-points. Examples of diatoms with inter-class similarity and intra-class variance are shown in Figure 4.1. Figure 4.1(a) shows the images of diatoms from three different classes that have similar appearance and are often confused by the classification network. Figure 4.1(b) shows the images from a single class, but taken from the side and top view. Here, the network fails to identify all of them as belonging to the same class.

An intuitive approach to tackle this problem would be to separate the feature embeddings of different classes by a distance so that the network can better differentiate them. This is known as metric-learning [31, 159] and the principle behind it is to bring closer the feature



(a) (b)

Figure 4.1: Examples of diatom images usually misclassified due to high inter-class similarity (a) or intra-class variance (b). (a) Three different diatom species of the genera *Nitzschia*: *N. soratensis*, *N. subacicularis* and *N. costei*, from left to right. (b) A single diatom species *Aulacoseira pusilla* seen from the side or from the top, from left to right.

embeddings of the objects belonging to the same class while pushing apart the embeddings of the objects belonging to different classes. This effectively minimizes the misclassifications due to inter-class similarity. To reduce the misclassifications due to intra-class variances, we develop a self-supervised representation learning method that automatically clusters the instances within the high variance classes. Then the generated clusters are considered as independent classes while training. The clustering is based on the learned visual features and so, each cluster contains similar feature embeddings. This way, the variance within each class is reduced, and the network learns fine-grained features specific to them. In addition, the algorithm automatically chooses the classes to be clustered and the optimal number of clusters to be generated. To demonstrate the generalizability of our method, we evaluate the performance on the Diatom Atlas dataset and the WHOI plankton dataset [133].

4.1.1 Related Works

Aquatic microorganism classification

Early methods of plankton and diatom classification relied on hand-crafted methods to extract the features for classification [44]. [171] uses invariant moment features and Fourier boundary descriptors to capture the shape and texture information of plankton images. [76] uses gray level covariance matrix to extract various textural descriptors, and Support Vector Machine (SVM) for classifying planktonic taxa. In [207], the classification performance is improved by extracting a range of morphological and textural features and using multiple kernel learning (MKL), which learns a convex combination of kernels to capture different aspects of the extracted features. [25] uses two feature extraction methods, SURF and LBP, to extract the features and applies PCA for dimensionality reduction to perform computa-

tionally efficient plankton classification. [19] analyses some commonly used morphological and statistical descriptors to classify the diatom images. Although the hand-crafted based methods have been proven to be successful in identification, it is time-consuming to choose the appropriate features. Thus, the recent focus is on using deep learning to extract features automatically.

One of the early methods applying deep learning for plankton classification, [36] trains a deep classifier with several augmentation methods to improve the zooplankton classification performance over the traditional hand-crafted methods. [145] develops an inception-based module to deal with multiscale images. Moreover, they place constrains on the capacity of each layer and the receptive field of the CNN to ensure improved performance when increasing the depth of the network. [119] compares the performance of various off-the-shelf deep neural networks for plankton and coral classification. Their experimental results show that using an ensemble of deep classifiers substantially improves the classification metrics over stand-alone classifiers. [103] uses transfer learning and downsamples the classes with large number of images to handle class imbalance in classification. [137] develops a combination of unsupervised and supervised learning to develop a plankton classifier that is robust to class imbalances and detect anomalies. Whereas, our work considers the influence of intra-class variance and inter-class similarity on the classification performance.

Dealing with inter-class similarity and intra-class variance

Few methods have been proposed in the computer vision literature to handle intra-class variance and inter-class similarity for visual recognition. [55] uses split-and-merge to handle high intra-image and intra-class variations for celiac disease diagnosis. [21] considers the instances contributing to high intra-class variance as outliers. They use triplet loss with a weighting scheme where each instance is given a weight based on how representative they are of their class. [142] uses a Hadamard layer to minimize the intra-class variance. The one closely related to ours is [49], where they cluster the instances within each class into a set of pre-defined sub-classes (K classes) using K-Means clustering. They calculate two sets of triplet losses: one for the broader class instances and the other for the sub-class instances. The drawback with this approach for our application is that since K-Means is applied to all the classes and K is pre-defined, it could result in over or under-clustering. Contrary to their approach, our algorithm uses X-Means [140] and decides the classes to be clustered and the optimal number of clusters during training. X-Means is a variant of K-Means that can automatically decide the number of clusters to be created based on certain conditions, such as the Bayesian information criterion (BIC) scores.

4.1.2 Method

The proposed method considers the two challenges of aquatic microorganism classification: (1) inter-class similarity and (2) intra-class variance. It uses a self-supervised regularization method, where the representations of the training samples are dynamically clustered using X-Means [140]. The samples are assigned new pseudo-class labels defined by their cluster assignment. The network is then optimized with the cross-entropy loss and the triplet loss, which tries to minimize the distance between the objects belonging to the same class while maximizing the distance between the objects belonging to different classes in the feature space. With periodic validation, the clusters are updated and the total number of classes change with every validation.

Training algorithm

This section describes the self-supervised automatic label refinement through clustering.

Notations. Consider a classification problem consisting of k classes with input samples $\boldsymbol{x}$ and labels $\boldsymbol{y} \in \{1, .., k\}$. The training dataset is denoted by $\boldsymbol{\mathcal{D}_{train}} = \{(x_n, y_n)\}_{n=1}^{N}$ and the validation dataset by $\boldsymbol{\mathcal{D}_{val}} = \{(x_m, y_m)\}_{m=N+1}^{M}$. Let the training input samples be represented as $\boldsymbol{x_{train}} = \{x_n\}_{n=1}^{N}$. The training is done with an off-the-shelf classification CNN. The penultimate layer of the CNN is used as a deep feature extractor $f^{\theta}(.)$, where $f^{\theta} : \mathbb{R}^D \rightarrow \mathbb{R}^d$ is a mapping from an input of dimension D to a representation (or feature) of dimension d, and θ is the model's parameters. The final layer consists of k neurons, followed by softmax activation to obtain the predictive probabilities.

Self-Supervised Dynamic Relabelling. During training, the training labels are updated to make them representative of the features' separation in the latent space. Every p epochs, the network is evaluated on $\boldsymbol{\mathcal{D}_{val}}$ and the classes with a false negative ratio higher than a threshold t are updated. This is representative of the scenarios where the samples of a given class are misclassified, which is typical of classes with high intra-class variations. For every class to update, the training representations belonging to such a class are extracted and clustered using X-Means [140]. The resulting clusters form well-separated groups and we use the cluster assigmnent as new pseudo-labels for the train samples. If k' additional clusters are introduced by X-Means, each of them are considered as independent classes. Thus, the number of classes becomes $K = k + k'$, and the final layer of the model is updated to have K neurons. Then, the network training continues with the new labels. During inference, the pseudo labels are remapped to the original set of k labels to identify their original class.

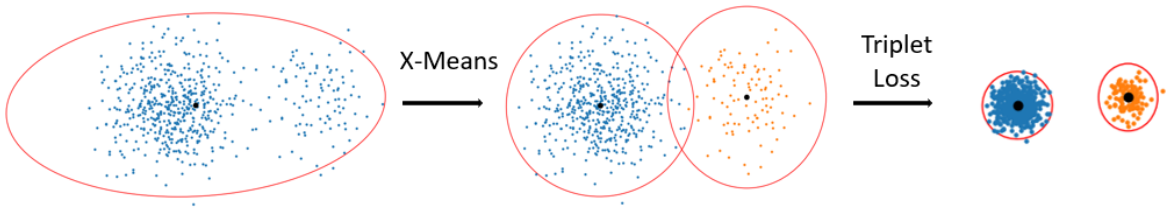


Figure 4.2: **Visualizing intra-class label refinement and feature optimization.** The original data is not perfectly Gaussian due to intra-class variations. X-Means refines the labelling by dividing the samples into multiple clusters that are approximately Gaussian. The clusters are considered as separate classes during training. Triplet loss optimizes the representations by bringing the in-class samples together and separating them from other classes.

Fig. 4.2 illustrates the benefits of jointly using X-Means and the triplet loss on the representations: X-Means splits classes with high intra-class variations into separated classes that are semantically more representative of the data, and the triplet loss reinforces this separation. The triplet loss can be formulated as follows: Let x_a, x_p and x_n be the anchor, positive and negative image. Here, x_a and x_p belong to the same class while x_n is from a different class. The triplet loss is given by

$$\mathcal{L}_{triplet} = \max\{||f(x_a) - f(x_p)|| - ||f(x_a) - f(x_n)|| + \alpha, 0\} \tag{4.1}$$

where α is the margin to separate the positive and negative images in the feature space. Our final loss is the sum of the cross-entropy loss (Eq. 2.8 defined in Chapter. 2) and the triplet loss.

$$\mathcal{L}_{total} = \mathcal{L}_{cross-entropy} + \mathcal{L}_{triplet} \tag{4.2}$$

The method introduces three hyperparameters: false negative ratio threshold t, frequency of validation epochs p and the maximum number of clusters (`max-clusters`), which is a parameter needed for X-Means.

Fig. 4.3 shows an example of clusters generated for the diatom class *Aulacoseira pusilla*. The diatoms have been clustered based on the difference in view-point and the porosity. Since they are considered as independent classes, the network learns these intricate details, which helps in fine-grained classification.

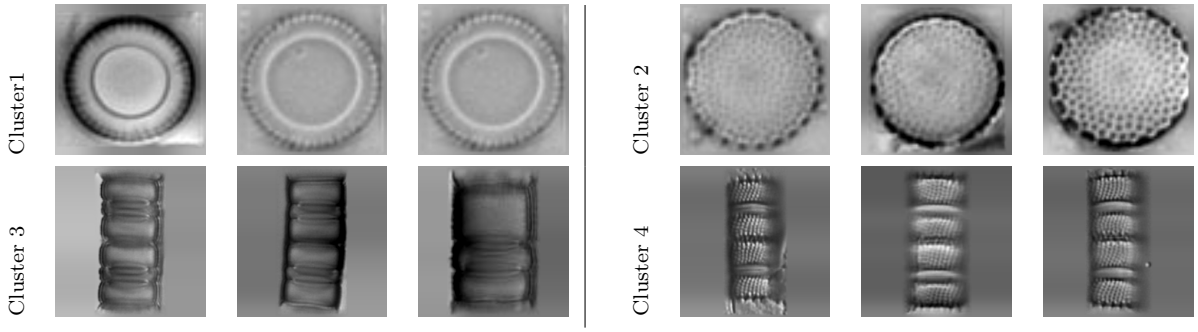


Figure 4.3: Clusters generated based on the learned visual features for the diatom class *Aulacoseira pusilla*.

4.1.3 Experiments

Datasets

We apply our method on two datasets:

Diatom Dataset The diatom dataset is a subset of the Atlas dataset described in Section 1.2.1. It consists of individual images of diatoms from the atlases of Rhône-Alpes [13], Île-de-France [96], and Bourgogne [139]. The dataset contains a total of 166 classes with a total of 9895 images.

WHOI-Plankton Dataset [133] The WHOI plankton dataset consists of 3.4 million images spread across 70 classes. We considered only those classes that have at least 50 images. Finally, we obtained 38 classes and a total of 26612 images.

All the images were padded and resized to size 256×256. We used K-Fold cross-validation with K=5.

Baselines

To evaluate our method, we perform experiments on the following baselines:

1. **Standard Classification** - We use a state-of-the-art classification network pre-trained on ImageNet dataset and fine-tuned to our dataset.

2. **Classification with triplet loss** - Along with the cross-entropy loss, we use the triplet loss. This method is done to study the impact of inter-class similarity on the classification performance.

3. **Classification with clustering** - This is our proposed clustering method, but using only the cross-entropy loss for classification. This is used to study the impact of intra-class variance on the classification performance.

4. **Classification with clustering and triplet loss** - This is our proposed method to minimize the impact of both the inter-class similarity and the intra-class variance.

5. **GS-TRS** [49] - This method uses K-Means to divide each class into K clusters and uses triplet loss for inter-cluster and inter-class objects.

We perform our experiments on two classification model architectures: ResNet50 [71] and EfficientNet [169].

Evaluation metrics

We evaluate the different approaches using the standard metrics used in classification, namely the classification accuracy, precision, recall, F-Score. The metrics are defined below:

- **Accuracy -** This is a measure of the overall correctness of a model's predictions. It is the ratio of correctly predicted instances to the total number of instances in the dataset.
- **Precision -** Precision measures the accuracy of positive predictions made by the model. It is the ratio of true positives to the total number of instances predicted as positive.
- **Recall -** Recall, also known as Sensitivity or True Positive Rate, measures the model's ability to identify all relevant instances in the dataset. It is the ratio of true positives to the total number of actual positive instances.
- **F-Score -** The F-Score is the harmonic mean of precision and recall. The formula is:

$$\text{F-Score} = \frac{2 * \text{Precision} * \text{Recall}}{\text{Precision} + \text{Recall}}$$

A higher value of these metrics indicates a better performance. Additionally, we also calculate the variance of the per-class false-negatives and false-positives. The variance gives us a measure of how consistent the classification is and so a lower value is preferred.

Implementation Details

We use Adam optimizer and the learning rate is 0.0002 and we use a batch size of 128. The output feature embedding dimension from the network is 256. We trained our networks on GeForce GTX 1080 with 12 GB RAM. 70% of the images were used for training, 20% for validation and 10% for testing.

Hyperparameter Tuning

Our training depends on the following hyperparameters: (1) **Frequency of epochs** p - After every p epochs, validation is performed to obtain the new cluster assignments using X-Means. (2) **False negative ratio threshold** t - t is a threshold used to decide the class features to be clustered. From the normalized confusion matrix obtained during the validation step, the classes having false negative greater than t are clustered using X-Means. (3) **Maximum number of clusters** (`max-clusters`) - This is a parameter of X-Means, that specifies the upper bound to the number of clusters that X-Means can generate for each class.

To find the optimal value of these parameters, a grid search was performed. For the grid search, the values of hyperparameters used were: $t \in \{0.0, 0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9, 1.0\}$, $p \in \{5, 10, 15, 20\}$ and `max-clusters` $\in \{3, 5, 7, 10\}$.

From the grid-search analysis, the best performance was obtained when $t = 0.3$, $p = 10$ and `max-clusters`=5.

4.1.4 Results

Table 4.1: Quantitative metrics for classification on the diatom dataset.

Architecture	Method	Accuracy	Recall	Precision	F Score	Variance	
						FN	FP
ResNet50	Std. Classification	94.24	93.54	93.98	92.85	0.036	0.049
	Classification+triplet loss	96.06	95.86	95.97	95.33	0.022	0.023
	Classification+clustering	96.57	96.53	96.61	96.15	0.020	0.014
	Ours	**97.37**	**97.95**	**97.97**	**97.79**	**0.0059**	**0.0071**
	GS-TRS [49]	93.23	93.03	93.44	92.66	0.018	0.018
EfficientNet	Std. Classification	95.31	94.92	95.10	94.82	0.031	0.041
	Classification+triplet loss	96.67	96.52	96.68	96.50	0.021	0.021
	Classification+clustering	96.76	96.61	96.11	96.67	0.019	0.016
	Ours	**97.22**	**96.64**	**97.30**	**96.69**	**0.0047**	**0.0053**
	GS-TRS [49]	93.60	92.96	93.27	93.43	0.020	0.024

Diatom dataset

Table 4.1 shows the quantitative metrics for the diatom classification. Results show that using both clustering and triplet loss outperforms the other methods. One interesting conclusion from the results is that classification with clustering performs better than classification with triplet loss. This means that intra-class variance has a higher impact on the classification performance than the inter-class similarity. Considering both the inter-class similarity and intra-class variance further improves the performance of the network. GS-TRS [49] is

not well-suited for this application because the number of clusters generated is not optimal and over or under-clustering deteriorates the performance.

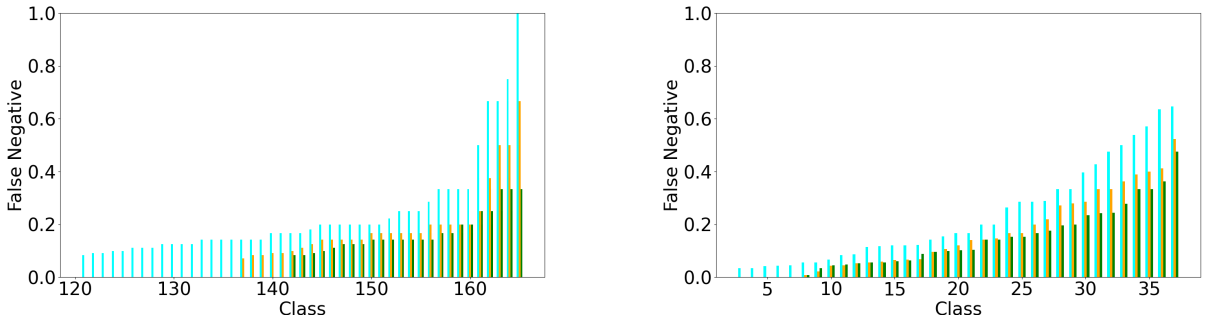


Figure 4.4: Per-class false negatives of standard classification (cyan), classification with clustering (orange) and classification with clustering and triplet loss (green) for (a) diatom dataset and (b) WHOI-Plankton dataset with EfficientNet. *Note: The classes with 0 false negatives are not shown here.*

For automatic classification of aquatic microorganisms, the certainty of prediction of a class is important, which means that the false negatives should be minimal. Fig. 4.4(a) shows a zoomed version of the false negatives with 3 methods: standard classification, classification with clustering, and classification with clustering and triplet loss overlaid onto a single graph. From the plots, 120 classes are perfectly classified when using a state-of-the-art classifier, whereas 140 classes are perfectly classified when using clustering with triplet loss. Also, the overall magnitude of the false negatives is reduced when using clustering and triplet loss than when compared to the other methods. This is a significant improvement, since the network can reliably be used to identify a larger number of classes than before.

WHOI-Plankton Dataset

Table 4.2 shows the quantitative metrics for classification on the WHOI-Plankton dataset. Similar to the diatoms, the clustering along with triplet loss outperforms the other methods. Figure 4.4(b) shows the overlay plot of the false negatives. In contrast to three classes that were perfectly identified by the state-of-the-art classifier, clustering and triplet loss improves it to seven perfectly identified classes. One could observe from Fig. 4.4(b) that when using only clustering and when using clustering along with triplet loss, the false negative magnitude does not change much. This could be due to the relatively low number of classes in the case of WHOI-plankton dataset and hence the inter-class similarity does not have a significant impact. One other important distinction between the two datasets is that the planktonic assemblage is at a higher hierarchy level of the aquatic microorganisms and encapsulates

Table 4.2: Quantitative metrics for classification on the WHOI-plankton dataset.

Architecture	Method	Accuracy	Recall	Precision	F Score	Variance	
						FN	FP
ResNet50	Std. Classification	88.54	84.90	85.82	84.71	0.022	0.136
	Classification+triplet loss	89.25	83.90	85.39	84.63	0.034	0.084
	Classification+clustering	88.17	**87.49**	83.25	84.52	**0.013**	0.095
	Ours	**89.48**	85.64	86.54	**85.50**	0.017	**0.034**
	GS-TRS [49]	87.53	78.75	**86.67**	80.81	0.035	0.156
EfficientNet	Std. Classification	88.82	86.51	82.21	81.67	0.030	0.125
	Classification+triplet loss	88.99	84.31	84.23	83.78	0.041	0.079
	Classification+clustering	88.71	85.96	81.65	81.49	0.015	0.103
	Ours	**90.53**	**88.91**	**87.13**	**83.86**	**0.013**	**0.024**
	GS-TRS [49]	87.66	82.25	82.35	82.03	0.033	0.147

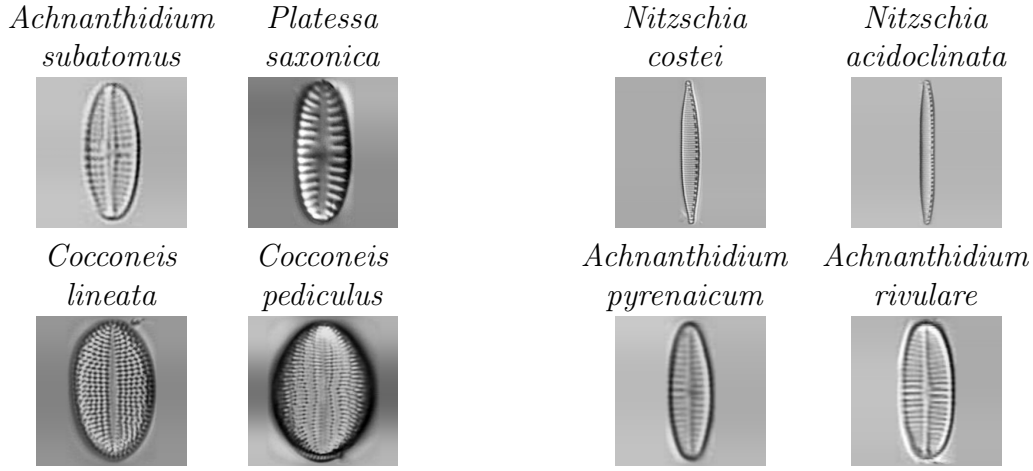


Figure 4.5: Pairs of classes most confused from the diatom dataset

wider variety of organisms. Whereas diatoms are a type of planktonic organisms and so are at a lower level of the hierarchy. This means that the diatoms are finer-grained and thus, one would expect them to have a higher inter-class similarity.

4.1.5 Discussion

We analyse the classes that were most often confused when using our proposed approach. Figure 4.5 and Figure 4.6 show some of the classes that were most confused, along with the class with which it was confused. From the results, one could see that these classes share high visual similarity.

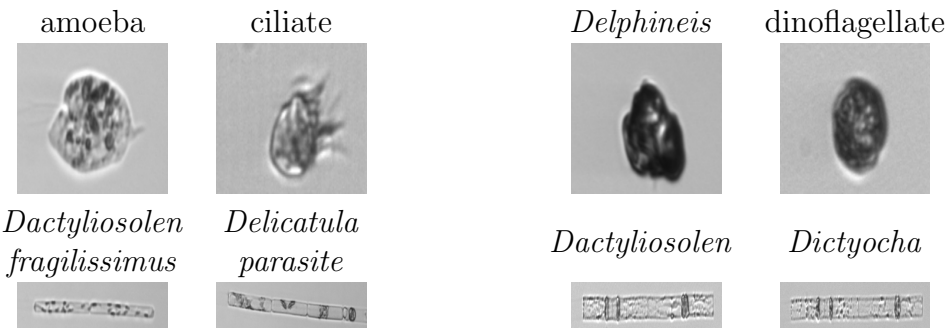


Figure 4.6: Pairs of classes most confused from the WHOI plankton dataset

4.1.6 Conclusion

In this section, we proposed a method to tackle the inter-class similarity and intra-class variance due to discrete image subsets, which is commonly found in microscopic images. Our method automatically identifies the classes to be clustered and the optimal number of clusters to be generated. Then these clusters are considered as independent classes while training a classification network. Finally, to deal with the inter-class similarity, we use triplet loss to separate out the features between each class. Using this approach, the network was able to learn finer-grained features that improved the classification performance. This was validated using quantitative metrics on the diatom and the WHOI-plankton dataset.

4.2 Uncertainty Estimation in Classification

In the previous section, we addressed a challenge in classifying microscopic images with objects exhibiting high inter-class similarity and intra-class variance. In this section, we address two difficulties commonly faced by most of the deep learning classification networks: (i) the derivation of calibrated classification *i.e.,* predict probabilities that represent true likelihood and, (ii) a measure of the classification uncertainty. Without those, the network makes incorrect predictions on the OOD data with high confidence [67] and no human-in-the-loop can catch such errors. This can lead to detrimental consequences, especially in safety-critical applications such as biology, robotics, and medical image analyses [66,111]. In the case of diatom classification, uncertainty estimation can help identify OOD data, that could arise due to the incorrect detections in the preceding stage, or when encountering new species that the network was not trained on. A calibrated network would assign a low prediction probability for these images, which translates to a high uncertainty in the network's prediction. This can help the users make informed decisions when using the predictions for further analyses. Another application of uncertainty estimation is in assisted

labelling/active learning. This is used to aid the experts in their labelling decisions. When new images are to be labelled by experts, the predictions are first obtained from the network. Then, the images are ranked according to the prediction probability obtained. This can be used by the labellers to prioritize their efforts on the less confident predictions.

Among deep uncertainty estimation approaches [4, 56, 58, 78] are the Bayesian Neural Networks [14], MC-Dropout [57] and Deep Ensemble [95]. These stochastic methods require multiple forward-passes, and are not scalable to large systems. Aware of the scalability requirements, current research focuses on estimating uncertainty from deterministic single-forward-pass networks [73, 114, 121, 144, 160, 175]. Distance-based methods belong to this category and are an attractive alternative for their excellent performance in OOD detection [104, 176].

Distance-based methods rely on the distance between the test samples and the In Distribution (ID) samples in a network's latent space to determine if the test samples are OOD. A relevant distance is the Mahalanobis Distance (MD) [120] for its superior performance over Euclidean Distance (ED) [84, 149, 178]. One key MD assumption though is that the in-distribution samples in the latent space should follow class-conditional Gaussian distributions. In practice, though, there is nothing in the classification training that constrains the latent space to fulfil such an assumption [40]. One reason for the non-Gaussianity is the high intra-class variance within the classes. These classes are usually composed of several clusters of visually similar images [22, 49]. This breaks the MD assumption, which could lead to incorrect or imprecise uncertainty estimation. The classification method, developed in the thesis for microscopic images, is applied in this situation. During training, the representations associated to a class that deviate from a Gaussian distribution are divided into several clusters that are approximately Gaussian using X-Means. The combination of in-class clustering and metric learning results in classification representations that are well-clustered and approximately Gaussian, which makes them suitable for MD-based uncertainty estimation. We call this method MAPLE , short for MAhalanobis distance based uncertainty Prediction for reLiablE classification. The code is available in: https://github.com/vaishwarya96/MAPLE-uncertainty-estimation.git

4.2.1 Related Works

Handling OOD data is an important challenge in the development of robust and reliable machine learning models, particularly in real-world applications where unexpected or novel inputs are likely to occur.

There are various approaches to handling OOD data in machine learning. One common technique is to use anomaly detection methods, where the network recognizes and flags inputs that deviate significantly from the training data [195]. [73] observed that maximum softmax probabilities of OOD data tend to be lower than those of the ID data. A drawback of using softmax probabilities is that they may not always be well-calibrated, meaning that they may not accurately reflect the true likelihood of the predicted class labels [67]. Hence, [109] uses temperature scaling for calibrating softmax probabilities, where the output probabilities are adjusted to better align with the true accuracy of the model. Additionally, they add noise to the inputs to improve the separation between the probabilities of ID and OOD data. [84,149] use Mahalanobis Distance (MD) between the test samples and training data in the classifier's latent space to detect the OOD samples. They rely on the principle that OOD data typically have a larger MD than the ID data. [192] uses contrastive learning during training, which pulls the in-class features together while pushing apart the inter-class features to enhance the OOD detection performance.

Another approach for OOD detection in machine learning is to use uncertainty estimates to identify when the network is uncertain about its predictions and flags such instances for further inspection [72]. In this thesis, we will develop methods using the latter approach, which is the focus of this section.

Uncertainties in the machine learning literature are divided into two main categories:

- **Epistemic Uncertainty** - Also known as model uncertainty or reducible uncertainty, refers to the uncertainty that arises from the limitations of the model itself, which can be reduced with additional data or improved models. It is associated with the lack of knowledge or understanding about the data-generating process or the underlying relationships between the inputs and outputs. Epistemic uncertainty can arise from various sources such as insufficient or biased training data, overfitting, model misspecification, and other factors that affect the model's ability to capture the true underlying patterns in the data.

- **Aleatoric Uncertainty** - This uncertainty is due to the inherent noise in the data. This can occur due to measurement errors, sensor noise, natural variation in data, and other stochastic factors that contribute to the randomness or unpredictability of the data. It is also known as irreducible uncertainty, since it cannot be reduced with

additional information or better models.

Both the epistemic and aleatoric uncertainty contribute towards the **predictive uncertainty**, which refers to the uncertainty in the predictions obtained from the model.

Multi-forward-pass Uncertainty Estimation

Traditional uncertainty quantification methods rely on Bayesian Neural Networks [63, 83] to learn a distribution over the network weights. To extract predictive probability variance, sampling [29] or variational methods [14] are used. The application of these methods is limited, as they increase the number of parameters by a factor of two and hinder convergence. As a lighter alternative, MC Dropout [57] enables dropout at test time and averages the network's output over several forward passes. While MC Dropout paves the way towards faster and lighter uncertainty estimation, it has been shown to produce over-confident predictions [95] and underestimate uncertainty [165]. To improve uncertainty estimation, Deep Ensembles [95] average the predictions from an ensemble of trained models and achieve state-of-the-art performance on several classification tasks. It remains computationally expensive due to the training of multiple models and the several forward passes during inference. By deriving uncertainty from a single forward pass, MAPLE achieves significantly faster inference time without sacrificing performance.

Single-forward-pass Uncertainty Estimation

One line of work relies on the distribution of data samples in the network's latent space. A test sample is considered ID if it lies within the training data manifold, otherwise it is labelled as OOD. Methods differ in the way they regularize the representation space and the way they derive distances. DUQ [176] uses a Radial Basis Function (RBF) kernel in the representation space to measure distances between test samples and the centroids of various classes. Additionally, they use gradient penalty to obtain a regularized space, which improves the prediction's quality. SNGP [112] uses Spectral Normalization on the network's weights to satisfy the bi-Lipchitz condition, which is a more gradient-friendly regularization than DUQ. This condition preserves semantically meaningful distance changes in the representation space with respect to input changes. The prediction's uncertainty is then given by a Gaussian Process layer on the output. To improve the scalability of the Gaussian Process estimation, [175] proposes Deep Kernel Learning to process the input images with a distance-preserving network and fit a Gaussian on inducing points only. Contrary to these methods, MAPLE avoids the Gaussian Process estimation and gradient regularization during training and instead relies on simple metric learning. Similarly, VMDLS [40] simplifies

the Gaussian enforcement by training the network with a KL-divergence loss so that each class representation follows an isotropic Gaussian distribution in the latent space. However, this ignores the possible intra-class variation within each class and requires the Gaussian variance to be tuned manually. Instead, MAPLE uses a simpler self-supervised clustering that automatically fits the data. Also, MAPLE makes the latent space not only suitable for OOD detection but also for calibrated probability prediction.

Mahalanobis-Distance for OOD detection

MD is a common distance in the OOD detection literature. Early work by Lee et al. [104] derives confidence values as a function of MD to predict the likelihood of a sample being ID. To obtain competitive performance, the method requires several tweaks such as adding noise to input samples, combining confidence values from multiple feature layers, and fine-tuning on OOD datasets. [84] proposes two light improvements: Partial MD and Marginal MD. In Partial MD, the MD is computed on lower dimensional representations with PCA. Marginal MD uses all training representations to fit a single Gaussian to calculate the MD. While both perform well on Far-OOD datasets *i.e.,* where ID and OOD samples are significantly distinct, their results are limited on Near-OOD [52], where the OOD samples are semantically similar to the ID ones. Relative MD (RMD) [149] improves the MD performance on Near-OOD by computing a global MD between the test sample and the samples of all classes combined, and then subtracting this value from the per-class MDs. All these methods exhibit satisfying performance, but their main limitation is their strong assumption that the image representations follow a Gaussian distribution, even though standard classification training does not enforce such a constraint. MAPLE addresses this limitation with a self-supervised regularization. By doing so, the features better fit the theoretical framework of MD-based OOD detection, thereby improving the performance.

4.2.2 Method

In this section, we describe MAPLE, a method for MD-based OOD detection, uncertainty estimation, and calibrated classification. The training method is akin to the procedure explained in Sec. 4.1.2. X-Means clusters the classes with non-Gaussian representations into multiple Gaussians. This facilitates the utilization of MD for detecting out-of-distribution (OOD) data. These clusters are assigned new pseudo-class labels based on the samples' cluster assignments. The network is trained with the cross-entropy loss and the triplet loss. With periodic validation, the clusters are updated and the total number of classes adapt until they follow Gaussian distribution.

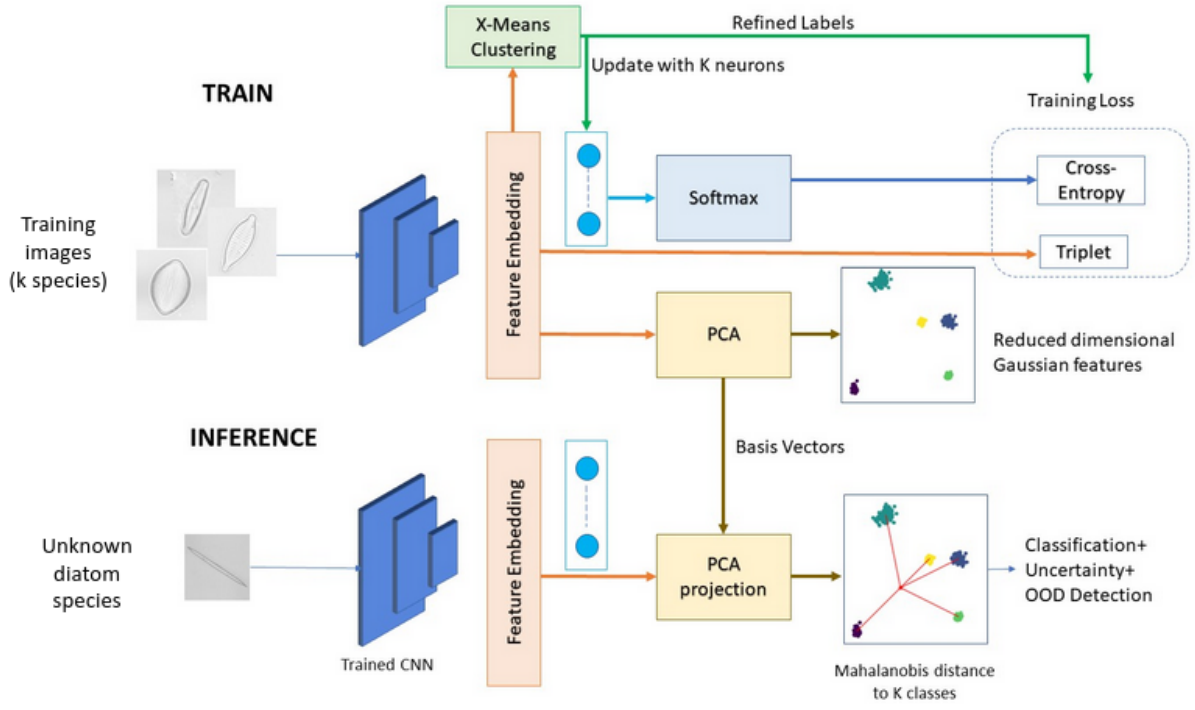


Figure 4.7: **Representation regularization with MAPLE for uncertainty estimation.** Our approach trains a classification network to learn representations that are approximately Gaussian for each class. During inference, the Mahalanobis distance between a test sample and the class centroids is used for classification, uncertainty estimation and OOD detection.

At inference time, the MD between a test sample and each cluster's centroid is used to estimate the classification uncertainty and the probability of the point being OOD. The pipeline for MAPLE is shown in Figure 4.7. Note that the only modification to the original network architecture is in the final layer, where the number of output neurons change according to the number of clusters identified. This makes MAPLE easy to integrate to any classification network.

Clustering. The motivation for using X-Means over other commonly used clustering methods such as K-Means [115], DB-SCAN [50] and Gaussian Mixture Models (GMMs) are two-folds: (1) X-Means is scalable and automatically identifies the number of clusters based on the Bayesian Information Criterion (BIC); (2) BIC uses a maximum likelihood estimation of the variance under the spherical Gaussian assumption, which means that the samples are approximately spherical Gaussian in each cluster.

Representation Distance

This section describes the MD derivation over the latent representations. To avoid matrix singularities, the latent representations are first reduced using PCA.

Dimensionality Reduction. Representations extracted from large neural networks usually have a high dimension and redundant dimensions. The MD requires calculating the inverse covariance matrix of these features, but the presence of redundancy causes the covariance matrix to be singular. Furthermore, [149] shows that the presence of non-informative dimensions could be detrimental to MD performance. This motivates the use of dimensionality reduction.

A common dimensionality reduction method is t-SNE [177], widely used for latent space's visualization. While t-SNE maintains the local distribution of points, it fails to represent global distributions accurately, which is undesirable in distance-based uncertainty predictions. Instead, we use Principal Component Analysis (PCA) for dimensionality reduction. The principal components are constructed from the covariance matrix of the standardized training representations. The eigen vectors of the covariance matrix are the principal components and the eigen values account for the amount of original information (variance) present in these components. We automatically estimate the number of principal components by the number of eigen values in decreasing order, required to explain 95% of the original data variance. This transformation is denoted by $g : \mathbb{R}^d \to \mathbb{R}^{d'}$, where d' is the dimension of the reduced features. With $\boldsymbol{x'_{train}} = f^{\theta}(\boldsymbol{x_{train}})$ the full dimensions training features, we denote $\boldsymbol{z_{train}} = g(\boldsymbol{x'_{train}})$ the reduced features.

Mahalanobis Distance. The MD is a generalized version of Euclidean distance that takes into account the data correlation to measure the distance. Hence, the MD is more accurate when predicting the distance between a point and a distribution of points. Here, MD is calculated on the PCA-reduced representations as follows. Let $\{z_i\}$ be the set of training representations after dimensionality reduction, μ_c be the class centroids with $c = 1, 2, ..., K$, and $\mathbf{\Sigma}$ be the shared covariance for all training samples, given by

$$\begin{aligned} \mu_c &= \frac{1}{N_c} \sum_{i:y_i=c} z_i \\ \mathbf{\Sigma} &= \frac{1}{N} \sum_{c} \sum_{i:y_i=K} (z_i - \mu_c)(z_i - \mu_c)^T \end{aligned} \tag{4.3}$$

The following Eq. 6.4 gives the Mahalanobis distance between the centroid μ_c of class c and a test sample $\tilde{x}$ with reduced representation $\tilde{z} = g(f^\theta(\tilde{x}))$

$$MD_c(\tilde{x}) = \sqrt{(\tilde{z} - \mu_c)^T \mathbf{\Sigma}^{-1} (\tilde{z} - \mu_c)} \tag{4.4}$$

Classification and Uncertainty Estimation

We now show how to use the MD distance calculated in Eq. 6.4 for three purposes: classification, predictive probability, and uncertainty prediction.

MD-based Classification. The predicted class is the one whose centroid c^* is closest to the test sample $\tilde{x}$:

$$c^* = \underset{c}{\text{argmin}}(MD_c(\tilde{x})) \tag{4.5}$$

Predictive Probability. We convert the MD into a calibrated classification probability using the following property: the squared MD on representations with dimension d' follows a chi-squared distribution $\chi^2_{d'}$ with d' degrees of freedom. The MD is converted as follows:

$$P^c_{MD} = 1 - \text{cdf}(\chi^2_{d'})(MD_c(\tilde{x})^2) \tag{4.6}$$

where cdf(.) is the cumulative distribution function. P^c_{MD} represents the probability that a test sample belongs to class c. When the test point belongs to a particular class, the MD to that class is low and the corresponding P^c_{MD} is high. The predictive probability is the one associated with the class c^* obtained in Eq. 4.5:

$$P^{c^*}_{MD} = \max_c(P^c_{MD}) \tag{4.7}$$

Note that contrary to the CNN softmax probability, this classification probability is calibrated and can be interpreted as a confidence in the classification output.

Uncertainty Prediction. We define the predictive uncertainty, which is the uncertainty in the network prediction as

$$u_{c^*} = 1 - P_{MD}^{c^*} \tag{4.8}$$

For small values of MD, u_{c^*} is around 0 and increases towards 1 as the MD increases.

4.2.3 Experiments on Diatom Dataset

We compare MAPLE with the following baselines: two multi forward-pass methods MC-Dropout [57] (10 dropout samples) and Deep ensemble [95] (10 models) and a single forward-pass method SNGP [112].

Evaluation Metrics

We report the standard evaluation metrics [112, 176] namely, the classification accuracy, the Expected Calibration Error (ECE), the Negative Log-Likelihood (NLL), the Area Under the Receiver Operating Characteristics (AUROC) and the Area Under the Precision-Recall curve (AUPR). For qualitative analysis, we use calibration plots. As mentioned previously, MAPLE produces two classification outputs, so we report the accuracies obtained from both the traditional softmax probability and the MD-based classification (Sec. 4.2.2). The ECE and the NLL are calculated from the predictive probability $P_{MD}^{c^*}$. AUROC and AUPR are calculated from the uncertainty u_{c^*}. The definition of these metrics are as follows:

- **Expected Calibration Error.** ECE is a measure of predictive probability calibration error. The output probability is divided into a histogram of B equally spaced bins. The expected calibration error gives the difference between the *observed relative frequency* (accuracy) and the *average predicted frequency* (confidence).

$$ECE = \sum_{b=1}^{B} \frac{n_b}{N} |acc(b) - conf(b)|$$

  where n_b is the number of samples in bin b, N is the total number of samples, $acc(b)$ and $conf(b)$ are the accuracy and confidence of bin b. A lower ECE score means that the accuracy and confidence are aligned, indicating better calibration.

- **Negative Log Likelihood.** NLL calculates the negative log-likelihood for the predicted class probability. While it is generally used for optimization using cross-entropy

loss, it is also commonly used to evaluate the prediction uncertainty. A lower NLL score is preferred.

- **Area Under Receiver Operating Characteristic Curve.** AUROC indicates the ability to separate ID and OOD samples. To calculate this metric, the predicted uncertainty is used to determine if a sample is ID or OOD. This can be considered as a binary classification problem. The area under the plot between the true positive rate and the false positive rate gives the AUROC value. Higher AUROC value means better separation between ID and OOD.

- **Area Under Precision-Recall Curve.** AUPR, like AUROC measures the ability to separate ID and OOD samples. Considering ID and OOD separation as a binary classification problem, the area under the plot between precision and recall values give the AUPR score.

Implementation Details

For the experiments, we use the subset of the Atlas dataset used in Sec. 4.1.3 and the UDE dataset. The Atlas dataset is divided into ID dataset consisting of 130 classes (7874 images) and the remaining 36 classes as OOD (2021 images). 70% of the ID images were used for training, 10% for validation and 20% for testing. While training, horizontal and vertical flips were used for data augmentation. The following data subsets were used: (i) D25: In-distribution dataset includes classes with number of samples ≥ 25, consisting of 238 classes. The images from the remaining 408 classes were used as OOD dataset (ii) D50: In-distribution dataset includes classes with number of samples ≥ 50, consisting of 176 classes. The images from the remaining 470 classes were used as OOD. For both D25 and D50, 70% of the images from the in-distribution datasets were used for training, 20% for validation and 10% for testing.

The network architecture is Wide ResNet 28-10 [200]. The feature embedding layer has a dimension of 640. After training, there were a total of 158 classes, hence the output layer consists of 158 neuron with a softmax activation. We trained the model for 100 epochs with an Adam optimizer [89]. The learning rate was $2e^{-4}$ and batch size 4. The training was performed on a 12Gb NVIDIA GeForce 1080Ti. The dimension of the features after PCA reduction was 31.

Results

Table 4.3: **Comparison of baselines and our method on Atlas Dataset.** With its top performance and state-of-the-art speed, MAPLE makes for a particularly applicable method for classification and OOD detection on real case datasets.

Method	Accuracy ↑	ECE ↓	AUROC ↑	AUPR ↑	Latency (ms/sample)↓
MC-Dropout [57]	0.936	0.039	0.548	0.589	129.7
Deep Ensemble [95]	**0.969**	**0.025**	0.589	0.570	146.81
SNGP [112]	0.954	0.196	0.798	0.826	26.25
MAPLE	0.963	0.036	**0.864**	**0.865**	**17.38**

Table 4.4: **Experiments on the UDE Dataset.** MAPLE achieves a better OOD discrimination performance than the simple deterministic method.

Dataset	Method	Accuracy ↑	AUROC ↑	AUPR ↑
D25	Softmax	**0.708**	0.661	0.675
D25	MAPLE	0.691	**0.739**	**0.743**
D50	Softmax	**0.729**	0.684	0.688
D50	MAPLE	0.706	**0.709**	**0.705**

Quantitative Evaluation. Table. 4.3 shows the metrics for the different baselines and our method on the Atlas dataset. MAPLE outperforms all baselines on OOD detection. While Deep Ensemble has a slightly better accuracy and ECE score, this occurs at the cost of the large computational resources required to train multiple models. In terms of the latency value, MAPLE significantly outperforms the other methods and has the least latency value, and runs approximately 8 times faster than Deep Ensemble. Table. 4.4 shows the metrics for the comparison of MAPLE with the classifier softmax probabilities. The softmax probability has a high accuracy but MAPLE achieves better OOD detection performance. The slight drop in the MAPLE accuracy could be due to the loss of information while performing PCA. However, it achieves better OOD detection performance when removing the redundant features (discussed in the next section, where we perform an ablation analysis of our method using standard computer vision datasets).

Qualitative Evaluation. Figure 4.8 illustrates the calibration plot generated by various methods. The calibration of MAPLE closely aligns with the ideal curve at higher confidence levels. Conversely, SNGP exhibits under-confidence and performs the poorest, as evident from both the qualitative plot and quantitative metrics. Notably, SNGP achieves the highest

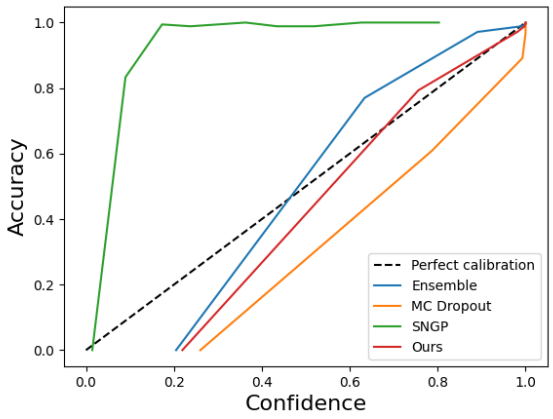


Figure 4.8: Calibration plot for the diatom dataset

ECE score. An explanation for this outcome can be attributed to SNGP's emphasis on preserving input distances in the latent space. However, this characteristic is disadvantageous for fine-grained datasets like diatoms, where high input similarities result in small distances. Consequently, the latent space exhibits small intra-class distances, with several overlapping classes. When using a Gaussian Process, the model is less confident in its predictions due to the overlapping feature representations, resulting in poor calibration. On the other hand, MAPLE employs triplet loss, which is designed to separate the representations of different classes in the feature space. This encourages larger distances between representations of different classes, making it easier to discriminate between visually similar classes using the MD, resulting in better calibration performance.

4.2.4 Experiments on Standard Datasets

We compare MAPLE with the following related works: two multi forward-pass methods MC-Dropout [57] (10 dropout samples) and Deep ensemble [95] (10 models), four single forward-pass methods: DUQ [176], SNGP [112], DUE [175] and VMDLS [40]. Following the standard evaluation on OOD detection, we evaluate the methods on classification, predictive probability calibration, and OOD detection on the three benchmark datasets: FashionMNIST [193] *vs.* MNIST [102], CIFAR10 [93] *vs.* SVHN [130], CIFAR10 *vs.* CIFAR100 [93].

We also compare MAPLE with MD-based methods on OOD detection, namely, the approach by Lee et al. [104], Marginal MD [84] and RMD [149]. We used the near-OOD CIFAR10 *vs.* CIFAR100 for the comparison, which is notably challenging for OOD detection.

Evaluation Metrics

We report the same evaluation metrics as the experiments on diatoms in Sec. 4.2.3. Additionally, we also plot uncertainty histograms for the qualitative analysis.

Implementation Details

FashionMNIST. FashionMNIST [193] consists of 10 classes. We split the original training set consisting of 60000 samples into train and validation set, in the ratio of 80:20. The validation set was used for hyperparameter tuning. The test set consists of 10,000 samples, which we used for inference and calculating the metrics. For analyses on OOD dataset, we use the test set of MNIST [102], containing 10,000 instances. For realistic evaluation, the normalization of MNIST is done the same way as FashionMNIST.

We use the network backbone of [176]. The CNN consists of three layers of convolution with 64, 128 and 128 3×3 filters, a dense layer for feature extraction and an output layer with softmax activation. Each convolutional layer is accompanied by a batch normalization and a 2×2 max pooling. The feature embedding's dimension is 256. The dimension of the final layer is equal to the total number of classes obtained after clustering, which was 14 for MAPLE . We trained the network for 50 epochs. For training, we used an SGD optimizer with a learning rate of 0.05, momentum 0.9, weight decay of $1e^{-4}$. The training was performed on a 12Gb NVIDIA GeForce GTX 1080Ti with a batch size of 128. The reduced dimensional feature after PCA had a dimension of 5.

CIFAR10. CIFAR10 [93] consists of 10 classes. We split the original training set consisting of 50000 samples into train and validation set, in the ratio of 80:20. The validation set was used for hyperparameter tuning. The test set consists of 10,000 samples, used for inference. For OOD analyses, we use the test set of SVHN and CIFAR100, which consists of 26,032 and 10,000 samples respectively. The OOD images are normalized the same way as train images during inference.

The network architecture is Wide ResNet 28-10 [200]. The feature embedding layer has a dimension of 640. After training MAPLE , the number of classes were 12, and hence, the final layer has a dimension of 12, followed by softmax. We trained the model for 200 epochs. We used an SGD optimizer with a learning rate of 0.05. The momentum was set to 0.9 and weight decay of $1e^{-4}$. The training was performed on a 12Gb NVIDIA GeForce GTX 1080Ti with a batch size of 64. The dimension of the reduced features from PCA is 12.

Results

We report the results on FashionMNIST and CIFAR10 in Tables 4.5 and 4.6 respectively.

Table 4.5: **FashionMNIST (ID) vs MNIST (OOD).** MAPLE achieves the best performance on OOD detection and has the best inference time. It is very competitive with other single-pass methods on the classification task. Blue: Classification based on prediction from softmax probability Orange: MD-based classification.

Method	ID metrics			OOD metrics		Latency↓
	Accuracy ↑	ECE ↓	NLL ↓	AUROC ↑	AUPR ↑	(ms/sample)
MC Dropout [57]	0.923	0.069	**0.213**	0.912	0.895	15.46
Deep ensemble [95]	**0.939**	0.018	0.238	0.874	0.866	23.87
DUQ [176]	0.923	0.045	0.276	0.941	0.945	2.61
SNGP [112]	0.924	**0.009**	0.259	0.981	0.978	2.54
DUE [175]	0.923	0.028	0.284	0.954	0.948	2.57
VMDLS [40]	0.920	-	-	0.963	0.970	2.60
MAPLE	0.925/0.924	0.020	0.262	**0.995**	**0.994**	**2.48**

Table 4.6: **CIFAR10 (ID) vs SVHN / CIFAR100 (OOD).** MAPLE outperforms all single and multi pass methods on OOD detection, and results in significantly faster derivation. Classification with MAPLE is very competitive with the state-of-the-art and the predicted probabilities are better calibrated. Blue: classification based on prediction from softmax probability. Orange: MD-based classification.

Method	ID metrics			OOD AUROC ↑		OOD AUPR ↑		Latency↓
	Accuracy ↑	ECE ↓	NLL ↓	SVHN	CIFAR100	SVHN	CIFAR100	(ms/sample)
MC Dropout [57]	0.960	0.048	0.293	0.932	0.835	0.965	0.829	27.10
Deep Ensemble [95]	**0.964**	0.014	**0.134**	0.934	0.864	0.935	0.885	38.10
DUQ [176]	0.945	0.023	0.222	0.927	0.872	0.973	0.833	8.68
SNGP [112]	0.957	0.016	0.153	0.991	0.911	0.994	0.907	6.25
DUE [175]	0.956	0.015	0.179	0.936	0.852	0.967	0.850	6.94
VMDLS [40]	0.951	-	-	0.932	0.868	0.953	0.864	5.61
MAPLE	0.956/0.954	**0.012**	0.142	**0.996**	**0.926**	**0.997**	**0.918**	**4.96**

OOD Detection Results. MAPLE outperforms all baseline methods by upto 12% on the AUROC and AUPR scores, and achieves so with the least computation time[1]. Note that competitive approaches, such as SNGP and DUE, derive their performance from spectral normalization and Gaussian process layer, which are invasive training add-ons. In contrast, MAPLE relies only on the layers of a standard CNN architecture to achieve superior performance.

[1]Latency value for MAPLE includes time for inference+post-processing with MD. Latency for MC Dropout and Deep Ensemble are when the inferences are performed serially.

When it comes to inference speed, MC Dropout and Deep Ensemble perform the worst, which is expected since they require multiple forward passes during inference. In contrast, most single-forward-pass methods achieve scores comparable to MC Dropout and Deep Ensemble while being faster, with a factor close to 8 times faster when comparing MAPLE and Deep Ensemble. This reinforces MAPLE's motivation: the distribution of feature points in a network's latent space holds reliable information for fast prediction of a network's uncertainty and detection of OOD samples.

Classification Results. MAPLE achieves results competitive to state-of-the-art, only 1% below the top method Deep ensemble [95] whose score comes at the cost of training and inference on several models. Note that both MAPLE accuracies, the softmax probability and the MD-based one are close. A finer analysis of the accuracy shows that the slight difference in accuracy with the MD-based classification occurs on samples the network is uncertain about: MAPLE achieves top accuracy on high-confidence predictions (above 80% and 90% confidence) and the accuracy slightly decreases for lower-confidence predictions. This is shown in Tables. 4.7 and 4.8 for the CIFAR10 and FashionMNIST datasets respectively. We evaluate the accuracy of prediction when selecting samples with predictive confidence above a given threshold. In other words, classification is performed only when the network's confidence is above a threshold. MAPLE achieves the best accuracy at confidence values of 0.80 and 0.90 on CIFAR10. Overall, on both CIFAR10 and FashionMNIST, MAPLE has competitive accuracy with the other approaches. This shows that even though MAPLE is computationally efficient, it can achieve the same level or better performance as the other methods.

Table 4.7: **Accuracy on CIFAR10 with different confidence levels.** MAPLE achieves top accuracy at confidence levels of 0.80 and 0.90.

Method	acc@.50	acc@.80	acc@.90
MC Dropout [57]	0.962	0.976	0.988
Deep ensemble [95]	**0.967**	0.987	**0.995**
DUQ [176]	0.950	0.977	0.982
SNGP [112]	0.959	0.978	0.985
DUE [175]	0.962	0.974	0.979
MAPLE	0.958	**0.989**	**0.995**

Calibration Results. MAPLE is competitive with state-of-the-art SNGP [112] and Deep Ensembles. When training on FashionMNIST, one source of ECE error is MAPLE's under-confidence on the accuracy range below 80%. This is visible in the calibration plot (Fig. 4.9) where the curve goes above the ideal calibration: the confidence is lower than the accuracy.

Table 4.8: **Accuracy for FashionMNIST samples with different confidence levels.** MAPLE achieves competitive accuracies at different confidence values.

Method	acc@.50	acc@.80	acc@.90
MC Dropout [57]	0.924	0.931	0.948
Deep ensemble [95]	**0.946**	**0.975**	**0.983**
DUQ [176]	0.925	0.947	0.962
SNGP [112]	0.931	0.963	0.977
DUE [175]	0.929	0.951	0.964
MAPLE	0.930	0.972	0.974

FashionMNIST CIFAR10

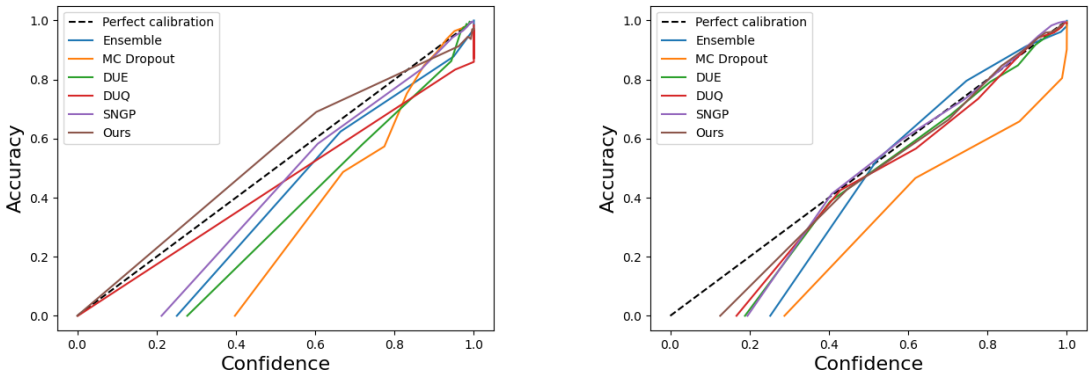


Figure 4.9: **Calibration plots.** A perfectly calibrated plot is when the predicted confidence equals the true likelihood *i.e.,* the accuracy. This is shown by the linear dotted line in the plots. MAPLE is closer to optimal calibration than existing methods, especially for low-accuracy samples.

This is typical of the scenario where the inter-class representations are widely spread out. Even though a sample falls closest to its ground-truth centroid, their inter-distance remains high, which decreases the confidence. The sample is then correctly classified, but with a low confidence. Note that while optimal calibration is the gold-standard, MAPLE's under-confidence still makes it more compliant with hazardous applications than other methods that make over-confident predictions, which can be disastrous. On CIFAR10, all methods are well-calibrated, except for the overconfident MC-Dropout, which explains its high ECE score. When the accuracy is below 0.4, baseline methods become overconfident whereas MAPLE is closer to optimal calibration and achieves the best ECE score.

Uncertainty Histograms. Uncertainty histograms (UH) are a means to visualize the uncertainty values predicted. When provided with OOD samples, it is expected that the network makes predictions with high uncertainty. The frequency of predicted uncertainties is plotted as a histogram. A high frequency at top uncertainty ranges show that the network

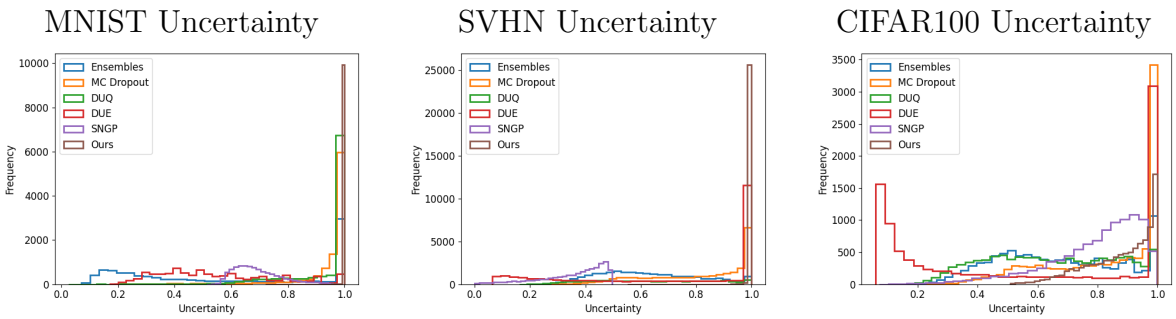


Figure 4.10: **Uncertainty Histograms for OOD datasets.** A high frequency of prediction at top uncertainty ranges indicate that the network is uncertain about its prediction. MAPLE outperforms the other methods and predicts OOD samples with high uncertainty. In other words, MAPLE is able to correctly identify these samples as OOD.

is uncertain when provided with OOD. Fig. 4.10 shows the uncertainty histograms for the different approaches.

From the plots, MAPLE assigns high uncertainty for OOD datasets. Compared to the other methods, MAPLE exhibits a higher frequency peak at an uncertainty around 1 for MNIST and SVHN. While MC-Dropout and DUE has a high frequency at uncertainty of 1 for CIFAR100, these methods also have a relatively higher peak at low uncertainties, since they make over-confident predictions on some OOD samples. Whereas, MAPLE's uncertainty values are spread across the higher end, which is desirable.

Gaussianity Test. In Section 4.2.2, it was theoretically shown that X-Means creates clusters of feature points that are Gaussian. In this section, we empirically test this. A commonly adopted method to check for multivariate Gaussian is to use a quantile-quantile plot, where an observed quantile is compared with a theoretical one. If the samples are Gaussian, their squared MD follows a χ^2 distribution. Thus, we use MD^2_{c*} of the samples feature embeddings as our observed quantile and compare with theoretical χ^2 quantiles.

For our test, we use the reduced feature embeddings, $\boldsymbol{z_{train}}$, from a standard classifier network and MAPLE . The MD^2_{c*} of samples are calculated and plotted with χ^2 quantiles with d' degrees of freedom, where d' is the dimension of feature embeddings. We measure the error, which is the mean absolute difference between the two quantiles, to test which method generates feature embeddings that are closer to a Gaussian. In the ideal situation, this value should be zero. The larger the error, the greater is the deviation from a Gaussian distribution. Table 4.9 shows the errors computed on feature embeddings from CIFAR10 and FashionMNIST dataset. From the results, MAPLE's error is reduced by over 50%, which shows that the feature representations of MAPLE are more Gaussian than when using a standard DNN classifier.

Table 4.9: **Mean absolute error between squared MD and χ^2 distribution.** The lower the error, the more Gaussian are the samples. MAPLE's training generates sample distributions that are approximately Gaussian, fitting with the theoretical framework for MD calculation.

Method	CIFAR10	FashionMNIST
Standard CNN	3.540	2.564
MAPLE	**1.395**	**1.215**

Comparison with other MD methods

Setup. MAPLE is compared against MD-based OOD detectors [84,104,149]. These methods are tailored for OOD detection, so we report the metric relevant to this task only for the sake of fairness. We report the AUROC score on the challenging near-OOD dataset CIFAR10 *vs.* CIFAR100. The experiments are done with a Wide ResNet 28-10 [200].

Table 4.10: **Comparison with MD-based OOD detection.** MAPLE performs significantly better in OOD detection than existing MD-based methods on the CIFAR10 *vs.* CIFAR100 setup. By enforcing the learned representations to follow a Gaussian distribution, MAPLE allows for distance derivations that are more semantically meaningful.

Method	AUROC ↑
Lee et al. [104]	0.893
Marginal MD [84]	0.838
RMD [149]	0.897
MAPLE	**0.926**

OOD Detection Results. MAPLE achieves top-performance on Near-OOD detection (Table. 4.10), which supports MAPLE's representation regularization. Note that the primary difference between MAPLE and the baselines is their lack of constraints on the latent representation. In contrast, we force the samples of every class to be Gaussian before calculating MD. Non-Gaussian samples lead to incorrect mean and covariance calculations, resulting in incorrect distance values. The error is more pronounced when the samples deviate from the Gaussian distribution by a large factor. This explains why the MD-based approaches under-perform compared to MAPLE on Near-OOD.

Ablation analysis

In this study, we assess how the different components of MAPLE impact its performance. We train a wide ResNet 28-10 [200] network on CIFAR10 and use SVHN and CIFAR100 as OOD datasets.

Table 4.11: **Ablation study.** We evaluate the influence of several MAPLE components. **PCA** (1 vs 2) results in a significant improvement of the OOD detection by discarding non-informative dimensions. The distances derived on these reduced features are better representative of the similarity between the input samples. The **MD** (2 vs 3) is better suited than ED for calibrated classification and OOD detection, which reiterates conclusions already found in previous works. The **triplet loss** (2 vs 4) improves both the accuracy and the OOD metrics by increasing the class separation. **Clustering** alone (2 vs 5) also contributes to a better separation of the classes, but the results are not as significant. The joint use of **triplet loss and clustering**, as done in MAPLE (6) achieves the best results on both classification and OOD detection. Note: #Eig refers to the number of principal components, whenever applicable.

Method	ID metrics			OOD metrics - SVHN		OOD metrics—CIFAR100		#Eig
	Softmax Accuracy ↑	MD-based Accuracy↑	ECE ↓	AUROC ↑	AUPR ↑	AUROC ↑	AUPR ↑	
DNN+MD (1)	0.950	0.943	0.086	0.752	0.762	0.583	0.564	-
DNN+PCA+MD (2)	0.950	0.946	0.053	0.855	0.839	0.813	0.859	12
DNN+PCA+ED (3)	0.950	0.943	0.105	0.829	0.804	0.734	0.765	12
DNN+Triplet+PCA+MD (4)	0.954	0.953	0.013	0.945	0.948	0.912	0.894	11
DNN+Clustering+PCA+MD (5)	0.947	0.945	0.032	0.922	0.908	0.811	0.815	12
MAPLE (6)	**0.956**	**0.954**	**0.012**	**0.996**	**0.997**	**0.926**	**0.930**	12

Dimensionality Reduction. We consider two scenarios: **(1) DNN+MD** - A baseline where a standard DNN is trained with the cross-entropy loss and with no feature regularization. The MD is computed on the raw features, and we add a value of $1e^{-20}$ to the diagonal elements [149] to avoid a singular covariance matrix. **(2) DNN+PCA+MD** - It follows (1) except that the MD is derived on PCA-reduced features.

Results: Dimensionality reduction (2) drastically improves the network's performance, as shown in the first line of Table 4.11. The improvement amounts to 7-30% on the OOD metrics and 3% on the ID metrics. One possible explanation is that the reduced dimensions are the ones that contribute to distinguishing ID samples from OOD ones, as previously observed by [149]. When including all the feature dimensions in the MD, the dimensions that do not contribute to discriminating ID and OOD samples add up and dominate the final MD score.

Distance Definition. We compare Mahalanobis distance and Euclidean distance (ED) in the network's latent space. We compare **(2) DNN+PCA+MD** with the new experiment **(3) DNN+PCA+ED** - It follows (2) except that the MD is replaced with ED. As for MD, the $\chi^2_{d'}$ distribution is used to obtain the probability values from ED (Sec 4.2.2).

Results: The results show that MD boosts the performance in terms of ID and OOD metrics. The improvement is ECE score is by 5%, and the OOD metrics improved by 3-9% when using MD. This is because MD takes into account the data correlation, which gives a better estimate of the probability and uncertainty values.

Representation training. To study the influence of the training on the representations, we consider three experiments: **(4) DNN+Triplet+PCA+MD** - We train the DNN using

both cross-entropy and triplet loss. **(5) DNN+Clustering+PCA+MD** - We train using the cross-entropy loss only and periodically cluster the feature points using X-Means. **(6) MAPLE** - This is our proposed method that fuses (4) and (5). For all experiments, the MD is derived on the reduced features.

Results: Using the triplet loss (4) improves the performance considerably compared to training with the cross-entropy loss only (2). An explanation is that the triplet loss pulls in-class feature embeddings together, and pushes the other class features apart. This encourages the representations to be well separated and makes it easier to distinguish OOD features. Choosing the triplet loss for metric learning is empirically motivated: experiments using contrastive loss showed that triplet loss has a slightly better performance.

Periodic clustering (5) improves the ECE score by 2%, and the AUROC and AUPR scores on SVHN by about 7% compared to (2). However, there is a slight drop in accuracy by 0.3% and OOD metric by 4% on CIFAR100. One explanation is that clustering increases the chances of new classes to overlap. This phenomenon is illustrated in the centre plot of Fig. 4.2. The class overlap is particularly hindering when the new domain is close to the training one: with clustering (5), the SHVN scores are better but the near-OOD CIFAR100 performs better without clustering (2).

MAPLE uses clustering together with triplet loss and achieves top-performance. The triplet loss reduces the overlap introduced with the clustering by pulling apart the newly created classes. With MAPLE, the latent representations are approximately Gaussian and well-clustered resulting in better MD estimates and superior performance in both ID and OOD metrics. Compared to experiment (2), the calibration error drops by 4% and the OOD scores improved by 4-11%.

False Negative Ratio t. We evaluate the influence of the clustering trigger *i.e.,* the False Negatives Ratio. We train MAPLE with a range of t values on CIFAR10 (Table 4.12).

Results: A low value of t results in overclustering, where multiple clusters contain similar images. This further increases the chances of misclassifications, leading to decrease in the metric values. On the other hand, high t values result in underclustering. Note that for $t > 0.3$, there are no additional clusters generated. This is because, the classes have false negative ratios that are below this threshold and so, they are not clustered. For CIFAR10, a t value of 0.3 yields the best results.

MAPLE evaluated on different backbones. MAPLE is tested on three networks: Wide ResNet 28-10 [200], ResNet-18 [71] and EfficientNet-B0 [169]. Table 4.13 gives the quantitative metrics for evaluation on CIFAR10 *vs.* SVHN and CIFAR100. While it is expected that the accuracy depends on the architecture used, the calibration and OOD de-

Table 4.12: **Metrics for different values of False Negative Ratio evaluated on CIFAR10** #Classes refers to the total number of output classes obtained after clustering. A low value of t results in overclustering, whereas a high t fails to detect classes with high variance.

False Negative Ratio (t)	#Classes	Accuracy↑	ECE↓	SVHN AUROC↑	CIFAR100 AUROC↑
0.0	23	0.9449	0.014	0.922	0.888
0.1	18	0.9534	0.013	0.964	0.918
0.2	14	**0.9544**	0.012	0.991	0.925
0.3	12	0.9541	**0.012**	**0.996**	**0.926**
0.4	10	0.9535	0.013	0.961	0.921
0.5	10	0.9535	0.012	0.955	0.915

tection are also influenced by the architecture. Wide ResNet, which has more number of parameters than the other two architectures, learns better feature representations for discriminating each class. As the model parameters decrease, there are overlapping feature points between different classes, which explains the lower accuracy and worse calibration and OOD metrics.

Table 4.13: **MAPLE evaluated on different architectures.** The metrics improve as the model parameters increase, suggesting that the network learns better discriminative feature representations, thereby improving the performance.

Architecture	Accuracy ↑	ECE ↓	SVHN AUROC ↑	CIFAR100 AUROC ↑
Wide ResNet 28-10 [200]	**0.954**	**0.012**	**0.996**	**0.926**
ResNet-18 [71]	0.945	0.029	0.979	0.886
EfficientNet-B0 [169]	0.902	0.035	0.942	0.893

Evaluation of different clustering methods. We analyse the performance of MAPLE when clustering is performed using K-Means, G-Means [206] and X-Means [140]. The value of K in K-Means is set to 3. Table 4.14 shows the results obtained. Based on the results, X-Means yields the best performance. K-Means and G-Means causes overclustering, which leads to worser performance on OOD detection. Using X-Means, we choose the optimal number of clusters, which performs superior to the others.

Effect of maximum number of clusters. Table 4.15 shows the results when the maximum number of clusters that can be generated for every class by X-Means is varied, along with different values of false negative ratio t for CIFAR10. For $t > 0.5$, none of the classes are clustered, and hence we do not include them. From the results, when the maximum

Table 4.14: **Metrics for different frequency of validation epoch** #Classes refers to the total number of output classes obtained after clustering. K-Means and G-Means lead to overclustering, whereas using X-Means, the optimal number of clusters are generated leading to better performance.

Clustering method	#Classes	Accuracy↑	ECE↓	SVHN AUROC↑	CIFAR100 AUROC↑
K-Means	30	0.952	0.154	0.871	0.850
G-Means	67	0.910	0.266	0.710	0.627
X-Means	12	**0.954**	**0.012**	**0.996**	**0.926**

Table 4.15: **Effect of maximum number of clusters per class on MAPLES's performance.** A high value of cluster numbers causes overclustering whereas a low value does not generate enough clusters. A value of 5 results in optimal number of clusters for MAPLE to learn meaningful representations.

Max. number of clusters	t	#Classes	Accuracy↑	ECE↓	SVHN AUROC↑	CIFAR100 AUROC↑
3	0.1	14	0.9542	0.012	0.996	0.925
	0.3	10	0.9540	0.014	0.972	0.919
	0.5	10	0.9533	0.012	0.958	0.917
5	0.1	18	0.9534	0.013	0.964	0.918
	0.3	12	0.9541	0.012	**0.996**	**0.926**
	0.5	10	0.9535	0.012	0.955	0.915
7	0.1	18	0.9537	0.013	0.959	0.894
	0.3	13	**0.9545**	0.012	0.992	0.921
	0.5	10	0.9531	0.013	0.944	0.911
10	0.1	26	0.9519	0.014	0.909	0.863
	0.3	22	0.9521	0.013	0.918	0.886
	0.5	11	0.9534	0.012	0.952	0.908

number of clusters is low, MAPLE fails to capture all the within-class variances, whereas higher values result in overclustering. With the maximum number of clusters as 5, MAPLE achieves the best performance.

Effect of frequency of validation epochs. Table 4.16 summarizes the metrics for CIFAR10 when the number of epochs after which the validation and cluster refinements are performed is varied. A low value of validation epochs does not give the network enough time to learn representations for the new clusters generated. Whereas, with larger number of epochs, the number of cluster refinements are low. In both these situations, the network does not identify the optimal clusters. MAPLE gives the best results when the validation is performed every 10 epochs.

Table 4.16: **Metrics for different frequency of validation epoch** #Classes refers to the total number of output classes obtained after clustering. With lower validation epochs, the clustering is too frequent for the network to learn meaningful representations. At lower frequency, the number of cluster refinements is not sufficient.

Validation epochs	#Classes	Accuracy↑	ECE↓	SVHN AUROC↑	CIFAR100 AUROC↑
5	16	0.895	0.025	0.914	0.876
10	12	0.954	0.012	**0.996**	**0.926**
15	12	**0.955**	0.012	0.987	0.922
20	10	0.953	0.013	0.968	0.917

4.2.5 Discussion

With the periodical clustering and the dynamic re-labeling, a natural question that arises is *'Is there a drop in performance when the ground truth labels change during training?'*. Experimentally, we observe a drop in training accuracy by 2-3% in the following epoch after every clustering phase. However, the network makes up for the drop within 4-5 epochs of training.

It can happen that the clusters contain very few samples, which introduces label imbalance when classifying. This is exacerbated when the samples are over-clustered. To mitigate this, we restrict X-Means to only cluster the classes that get misclassified. These are the classes with a false negative ratio higher than the threshold t. Automatic clustering regularization [23, 82, 99] is left for future work.

4.2.6 Conclusion

In this section, we presented MAPLE, a method for uncertainty estimation and out-of-distribution detection on CNN classifiers. The uncertainty is derived from the Mahalanobis Distance (MD) between an image representation and the class representations in the network's latent space. MAPLE derives meaningful MD distances by introducing a regularizer based on self-supervised label refinement and metric learning. Thus, MAPLE learns well-clustered representations that are approximately Gaussian for each class, which complies with the theoretical requirements of MD-based uncertainty estimation. Experimental results show that it achieves state-of-the-art results on out-of-distribution detection with the shortest inference time, and is very competitive with existing methods on predictive probability calibration. MAPLE also has the significant advantage of introducing the least architectural changes. Finally, we demonstrated the generalizability of our method by evaluating the performance on standard computer vision datasets.

4.3 Conclusion on Diatom Classification

In this chapter, we addressed the challenging task of diatom classification, which is hindered by both high inter-class similarity and significant intra-class variances. To overcome these obstacles, we developed a method that combines metric learning and self-supervised representation learning. This approach offers two significant advantages: Firstly, our method effectively separates the feature representations between each diatom class, resulting in distinct and discriminative class features. As a result, the classification performance is significantly enhanced, leading to a reduction in false negatives compared to standard classifiers. Secondly, the resulting latent space followed class-conditional Gaussian, enabling reliable uncertainty estimation through the application of the Mahalanobis distance. The probabilities estimated were calibrated, which greatly improved our ability to detect out-of-distribution samples.

Chapter 5

Misclassification Analysis in BDI

In Chapters 3 and 4, we created the essential tools for automating the detection and recognition of diatoms, which are crucial for bio-monitoring purposes. In this chapter, we will use the distribution of the identified diatoms to estimate the Biological Diatom Index (BDI) for biomonitoring. BDI is a widely used metric for evaluating water quality in French streams and rivers within the European Water Framework Directive. Accurate taxonomic identification of benthic diatom species is crucial to calculate BDI scores. However, misclassifications can impact the accuracy of the scores. In this chapter, we investigate the impact of misclassifications on BDI score and the related ecological status allocation. This work in under preparation for submission to a journal (Venkataramanan et al. cf Section 1.4).

5.1 Introduction

Human-induced changes are increasingly impairing the ecological quality of water bodies, leading to a loss in their biodiversity [106], ecological functions [201] and ecosystem services [146]. To clearly identify and address the sources of degradation, the European Water Framework Directive (WFD; European Council, 2000) has required to evaluate water body status by surveying some physicochemical and hydromorphological parameters, but also key biological indicators (= biological quality elements: BQEs). The main objective of the BQE-based assessment of a water body is to assign it a quality class (from "High" to "Bad") summarizing its ecological status, based on a robust biotic index that should consider the taxonomic composition and taxon abundances of the monitored BQE (European Council, 2000). Benthic diatoms are one of the BQEs recognized by the WFD for evaluating the ecological status of streams and rivers. In France, the Biological Diatom Index (BDI) [35] is a national standard routinely used to assess the quality of freshwater courses, based on the known pollution sensitivity, or "ecological profile", of 838 key diatom species. The ecological

profiles have been described through the species presence probability values along an ecological quality gradient of seven successive quality classes [35]. They are used to calculate the BDI score and determine the quality of watercourses.

For this purpose, diatom assemblages on benthic substrates (preferably coarse mineral substrates like pebbles and cobbles) are sampled according to a standardized method (NF T90-354, AFNOR 2016), processed and observed under the microscope to identify the species present. Even if the results of WFD-based monitoring programmes have to be provided with estimates of their levels of confidence and precision (European Council, 2000; Annex V §1.3), rather few studies have examined the sources of uncertainty related to the successive steps, from in situ sampling to the index-based final evaluation of the ecological status of the water body [20, 32, 33, 43, 68, 98, 124, 190, 191, 209]. However, this aspect is crucial to ensure both reliability in the ecological assessment of water bodies and effectiveness in management decisions for the conservation and restoration of aquatic ecosystems [152, 173]. Recently, a survey was conducted by [185] on the uncertainty in BDI due to inter-operator variability in field sampling and identifying the diatom species on morphological criteria. The results obtained from three expert operators showed a mean deviation of 0.85 BDI points (the BDI score is on 20 points) and a maximum deviation of 6.6 points. While this appears to be a low uncertainty value, further analysis revealed that the assignment to an ecological status class differed for about 45% of the study sites. This phenomenon occurs because many stations exhibited EQR scores that were not precisely aligned with the central area of a specific quality class but rather laid in proximity to a boundary between two classes, particularly between the "Moderate" and "Good" quality classes.

The taxonomic identification of diatoms is based on the morphological features exhibited by the diatom external silica shell (morphospecies), a step known to be subject to intra and inter-operator errors. One factor contributing to this is the dynamic nature of diatom taxonomy, which continually evolves over time [105]. The identification of new species based on morphological criteria can be challenging to apply consistently, leading to disagreements among experts in the field of diatom classification. This has motivated the automation of the identification step to improve the robustness of this crucial step in water body ecological status allocation, by minimizing the human biases, an ongoing challenge since the 90's [44].

With the increasing number of approaches utilizing deep learning-based classification for automation, we examine the uncertainties in the BDI scores due to the misclassifications from a CNN classifier. To illustrate this, we performed a simulation-based study on the influence of misclassifications on the hypothetical BDI scores, expressed as "Ecological Quality Ratio" (EQR), i.e. the ratio representing the relationship between the raw score of the biological index observed for a given water body and the value for this index in the reference

conditions applicable to that body (European Council, 2000). We generated thousands of synthetic diatom inventories representative of the Rhin-Meuse basin, the main hydrological basin in North-Eastern France, and calculated the corresponding EQR scores of the BDI index. We also analysed the robustness of the classifier to perturbations in the input images. The perturbations were chosen to closely mimic the real-life situations where the diatom images are subjected to random noise, occlusions, and blurring. In such conditions, it might be possible to identify the individuals accurately only at higher taxonomic levels (e.g. genus), since individuals at higher hierarchical levels share more common morphological features. Identifying the individuals accurately at higher hierarchical levels is beneficial, since it helps narrow down the potential set of species that an observed individual diatom could belong to. Motivated by this argument, we performed a sensitivity analysis on the robustness of the network at different hierarchical levels to correctly identify the individuals subject to perturbations. Finally, we performed an analysis to evaluate the influence of misclassifications at different taxonomic levels on the error on (i) BDI scores expressed as EQR and (ii) ecological status allocated to water bodies expressed in terms of water quality class.

5.2 Material and Methods

The overall pipeline adopted in this work is illustrated in Figure 5.1 and Figure 5.2. It is constructed from the classification framework presented in Section 4.1.2.

5.2.1 Generation of the synthetic diatom inventory dataset

BDI calculation requires the knowledge of (i) the diatom taxa present in the field sample, (ii) the relative abundance of each taxon in the sample, and (iii) the probability of occurrence of individual key species (838 key taxa listed in [35]) in each water quality class and its ecological amplitude. The values of the latter point are available in the BDI standard (AFNOR, 2016). To obtain the diatom taxa present and their abundances, microscopy images were acquired from the collected field samples. The first 400 diatom valves in the images were identified, and the relative abundance was calculated for each diatom taxon present. This information was available as inventories (thereafter called "real inventories"), which listed the diatom species and the abundance of each species in each standardized sampling event. Such inventories are available in the Naïades database (at `https://naiades.eaufrance.fr/`) gathering abiotic and biotic information on the quality of surface waters in France, and already described in [5,97]. The real inventory dataset consisted of 931 standardized sampling events performed over a span of nine years from 2005 to 2013, in the Rhin-Meuse basin. A total of 455 different

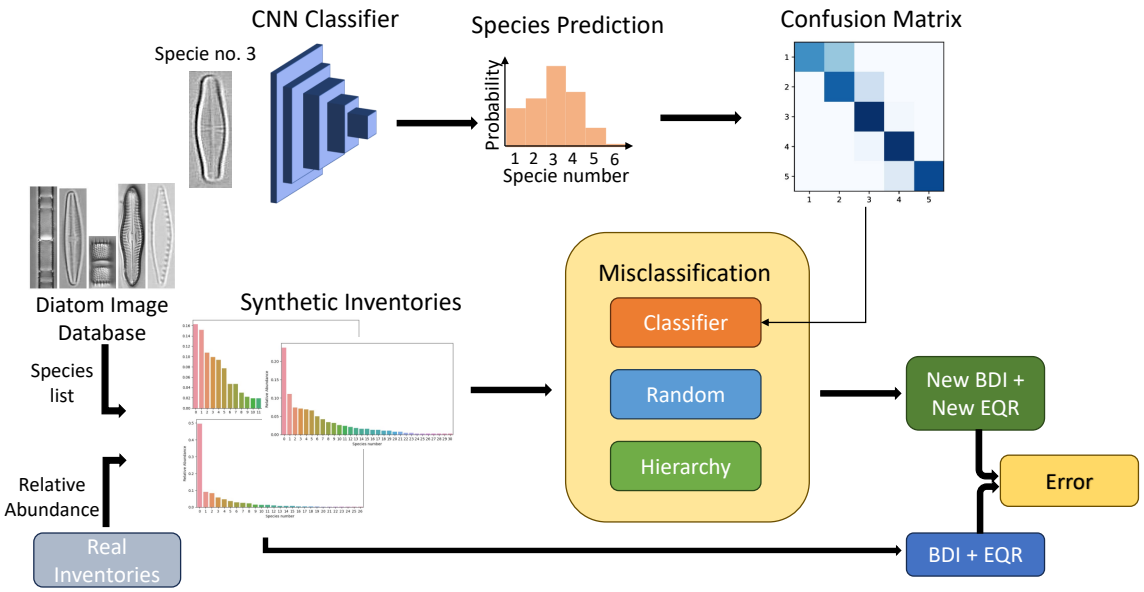


Figure 5.1: Visualization of the scheme for the BDI error propagation study. A CNN classifier is trained on the available image dataset, and the confusion matrix is available from its predictions. Synthetic inventories are generated using the relative abundances of the real inventories and the species from the available image database. Misclassifications are introduced in the synthetic inventories to estimate the error or uncertainty in the BDI and EQR scores.

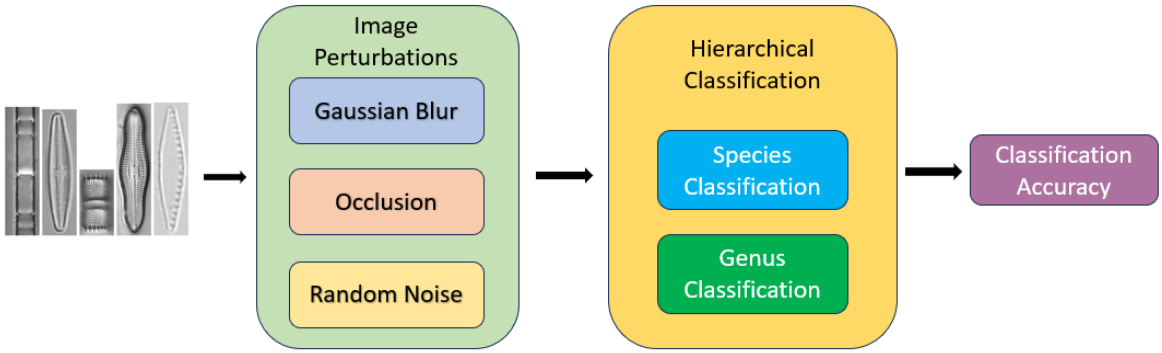


Figure 5.2: Visualization of the scheme for robustness analysis. First, the diatom images that are correctly classified by the network were perturbed using Gaussian blur, occlusion and random noise. The accuracy of the CNN classifier was compared on two levels of diatom taxonomic hierarchy: species level and genus level.

diatom species were identified over the whole set of real inventories. The aim of the following step was to generate synthetic inventories in order i) to maximize the coverage of species distributions in diatom assemblages and ii) to take into account the fact that among the 455 different diatom species which were identified over the 931 real inventories, a sufficient number of images was available for only 197 species to train the deep learning classifier network (see Section 1.2.1 on Atlas dataset).

To simulate inventories that were representative of the original ones, it was important that the 197 species used for simulation were ecologically representative of the 455 ones in the original data. With this goal, we analysed the array gathering the presence probability values of the 455 diatom species in the seven successive classes describing the ecological quality gradient in BDI by Correspondence Analysis (CA), and examined the locations of the 197 ones along factorial axes to evaluate their ecological representativeness of the full diatom assemblage (raw data availbale in [35]). The statistical analyses were performed using R (version 4.2.2).

Then the distribution of taxon abundances in each real inventory was used to generate synthetic inventories. To do this, for every real inventory the taxa represented by at least one individual were randomly replaced by taxa for which images were available in the image database. For every real inventory, the replacement of the taxon names was performed 100 times. Thus, with 931 real inventories, we generated a total of 93,100 synthetic ones, corresponding to real community structures.

5.2.2 BDI and EQR calculations

The BDI was calculated following [35]. The relative abundance of a species in a given sample was calculated by dividing its raw abundance by the sum of species abundances in this sample. The ecological profiles of diatom key species contain information about (i) their presence probability in the 7 successive classes describing the water quality gradient and (ii) the ecological amplitude V_x of this distribution. Let A_x be the relative abundance of every taxon x. The presence probability of x for class i is given as $P_x(i)$. Here, i = 1,2,...,7 represents the water quality class number. Assuming that there are N diatom species in the sampling event, the combined presence probability of the diatom assemblage corresponding to a given inventory, per quality class, is given by:

$$F(i) = \frac{\sum_{x=1}^{N} A_x P_x(i) V_x}{\sum_{x=1}^{N} A_x V_x} \tag{5.1}$$

The values of F are weighted – with the higher the quality class, the greater the weight – to obtain the B metric value as given below:

$$B = 1F(1) + 2F(2) + 3F(3) + 4F(4) + 5F(5) + 6F(6) + 7F(7) \tag{5.2}$$

The final BDI score is calculated from B using the rules described in Table 5.1.

Table 5.1: Rules for calculating the BDI score from the metric B.

B	$\leq$2	>2-6	$\geq$6
BDI	1.0	$(4.75B - 8.5)$	20.0

The BDI always has a value between 1 and 20, both included. To determine the Ecological Quality Ratio (EQR) for a specific river type, the BDI score obtained was compared to a reference value. This comparison was done by linking the BDI score to the appropriate river type reference value, following the guidelines specified in the JORF (Journal Officiel de la République Française, 2015) and [135]. The resulting EQR scores were then compared to the predefined class boundaries set by regulations to assess the ecological status of the site. The EQR score is given by:

$$EQR = \frac{\text{BDI} - \text{Minimum BDI observed}}{\text{Maximum BDI observed} - \text{Minimum BDI observed}} \tag{5.3}$$

Note that the minimum and maximum values are adapted to stream types, defined according to their size (5 categories from very small to very large) and their hydroecoregion (i.e. 22 distinct zones in mainland France, homogeneous in terms of geology, relief, and climate cf. [24, 189]). The ecological status class is obtained from the EQR score. The EQR threshold value for every class is given in Table 5.2.

Table 5.2: Threshold values on the BDI EQR for determining the ecological status (5 successive classes) of sites, where streams are draining a watershed of less than 10,000 km^2.

High/Good	Good/Moderate	Moderate/Poor	Poor/Bad
0.94	0.78	0.55	0.30

5.2.3 Deep learning classifier network

To train a deep learning classifier network, the Atlas dataset with 197 classes, described in Section 1.2 was used. Using this dataset, a deep learning classifier, with EfficientNet [169] as the backbone, was trained using PyTorch for classifying the individual diatom images using the same training parameters as Section 4.1.3. The network was trained on GeForce RTX 3090 with 24 Gb RAM. The classifier was evaluated based on the accuracy score.

5.2.4 Robustness analysis of the deep learning classifier

We analysed the robustness of the deep classifier network to perturbations in the data input and the effect of using hierarchical classification. First, the diatom images that are correctly classified by the network were perturbed. Then, the performance of the classifier on the perturbed images was used to estimate the robustness of the network. The performance was compared on two levels of diatom taxonomic hierarchy: species level and genus level. We consider three types of perturbations: (i) Gaussian blurring to blur the image, (ii) occlusion, where random parts of the image are replaced with black pixels, and (iii) addition of random noise to the input. These perturbations commonly occur when acquiring diatom images, and thus, it was important to consider the classifier's performance when provided with these types of images. We performed a sensitivity analysis, where for different levels of perturbation, we calculated the accuracy of the classification. The following parameters and values were considered for the sensitivity analysis:

1. Gaussian Blur - Gaussian blurring with kernel values of {1,3,5,7} are applied to the input image. The small kernel values (1 and 3) result in lighter blurring, preserving more fine details, while larger kernel values (5 and 7) lead to stronger blurring, smoothing out more pronounced features.

2. Occlusion – Rectangular portions of the input image are randomly erased by replacing the pixel values of the selected regions with 0. The range of aspect ratio of the erased region is 0.3-3.3. This variability ensures that the model encounters erased regions with different proportions, which helps it learn to handle various occlusion scenarios in real-world situations. The scale of the erased area with respect to the input image dimensions are in the range (0.02, 0.33). This range ensures that the occluded regions are not too small, as very small occlusions may not significantly affect the model's ability to learn robust features. Similarly, overly large occlusions could dominate the image and hinder the model's ability to extract meaningful information. The chosen scale range strikes a balance between maintaining the integrity of the input image and introducing significant occlusion for the model to learn from. The probability of random erasing are from the following six values: {0.0, 0.2, 0.4, 0.6, 0.8, 1.0}.

3. Random noise – Gaussian noise with 0 mean and standard deviation of 0.1 is applied to the input image. The probability of application of noise are from the following six values: {0.0, 0.2, 0.4, 0.6, 0.8, 1.0}.

5.2.5 Misclassification analysis

To analyse the effect of misclassification on the BDI values, we considered 3 scenarios:

1. CNN Classifier—Misclassification was introduced based on the confusion matrix obtained from a flat classifier. A confusion matrix summarizes how well a classification model predicts different categories, providing information about the model's accuracy and errors. It provides details on the ground truth and the predicted species by the network. This means when a species is misclassified, we can obtain the misclassified species and the fraction of individuals that were misclassified (probability of misclassification). Thus, when the network identifies a species with 100% accuracy, it means that the network never misclassifies it. Hence, in our error analysis, the species is not misclassified. For the others, the species to be misclassified and the probability of misclassification is obtained from a normalized confusion matrix.

2. Random—Misclassification was introduced by randomly replacing one taxon with another, i.e. regardless of the classifier performance. For every simulation, the taxon that was used to replace the existing taxon was chosen at random from the 197 taxa. This situation tries to represent the worst case scenario of misclassification.

3. Hierarchical Misclassification—While the 'Random Misclassification' introduces misclassifications randomly at the species level of the hierarchy, in this case, a species was confused with other species belonging to the same genus.

For the analyses, we considered misclassification rates from 10% to 100%, with successive steps of 10%. The misclassification rate tells the percentage of individuals from the inventory that will be replaced by others. For instance, if an inventory consists of 400 individuals and a misclassification rate of 10% is introduced, this means 40 individuals of the diatom assemblage will be replaced by individuals of other taxa. For the experiments, misclassification was introduced 50 times for every synthetic inventory. From the 93,100 synthetic inventories, a total of 4,655,000 misclassified inventories were generated.

5.3 Results

5.3.1 Representativeness of the subset of species available for simulating new inventories

Figure 5.3 shows the results of the CA applied to the [455 species x 7 water quality classes]. On the factorial planes defined by axes 1 and 2 (upper part), and axes 2 and 3 (lower part), the blue points and labels represent the locations of the 258 species present in the real inventories without sufficient number of images in open access atlases to be used for misclassification simulations. The 197 species (and their labels) used for simulating inventories (i.e. with at least 30 images available in open access atlases) were plotted in green. On visualizing the plots, it can be seen that species used for simulation homogeneously covered the entire ecological distribution of the original set of species. This suggests that the species used for simulations were satisfactorily representative of the ecological distribution of the whole taxonomic list of the 931 real inventories, and could be effectively used to generate synthetic inventories.

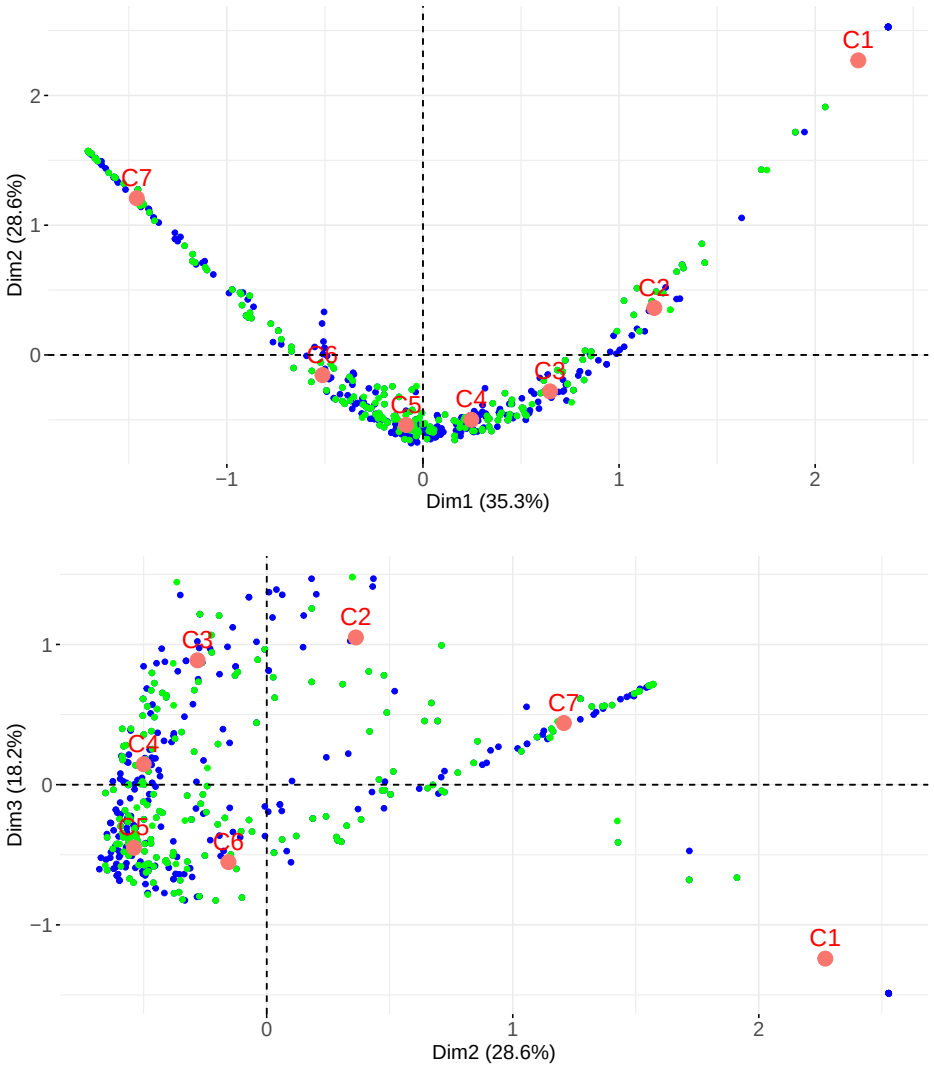


Figure 5.3: Ordination of diatom species by Correspondence Analysis (CA) based on their probabilities of presence in 7 water quality classes used to calculate the BDI. Locations of species in the factorial planes respectively defined by axes 1 and 2 (upper plot), and axes 2 and 3 (lower plot). The small points represent the distribution of the 455 species identified in the real inventory dataset: The green points represent the 197 species used to simulate the inventories (i.e for which we had at least 30 images) while the blue points represent the remaining ones (i.e. 258 species for which we did not have enough images). The red points represent the locations of water quality classes (C1 to C7, from the poor to the best water quality) at the weighted (per occurrence probability) average of the locations of taxa using these quality classes.

5.3.2 Classifier Performance

The classification network achieved an accuracy of 93.66%. Of the 197 species, 113 were classified with 100% accuracy. The remaining 84 species were misclassified, with a minimum rate of 1.25% to a maximum of 100% and a mean of 6.33% (Note: rate of misclassification = uncertainty = 100% - accuracy). The box plot of the misclassification rates is shown in Figure 5.4. The maximum value in the data range was 21.27%. Figure 5.5 shows images of the species that were misclassified by the classifier. The full list of misclassified taxa and the misclassification rates are provided in Table C.1 of Appendix C.

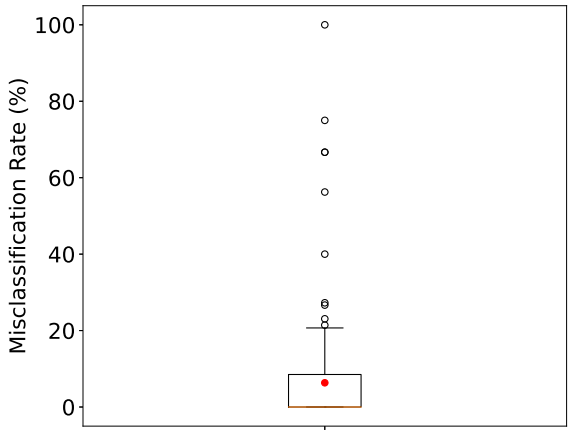


Figure 5.4: Box plot of the misclassification rates (in %) of the classifier. The box represents the interquartile range (IQR), the whiskers represent the minimum and maximum range in the data. The red point shows the mean value (n = 197), and the orange line shows the median value. The points outside the whiskers are the outliers *i.e.,* corresponding to species outside the +/- 1.5 interquartile range.

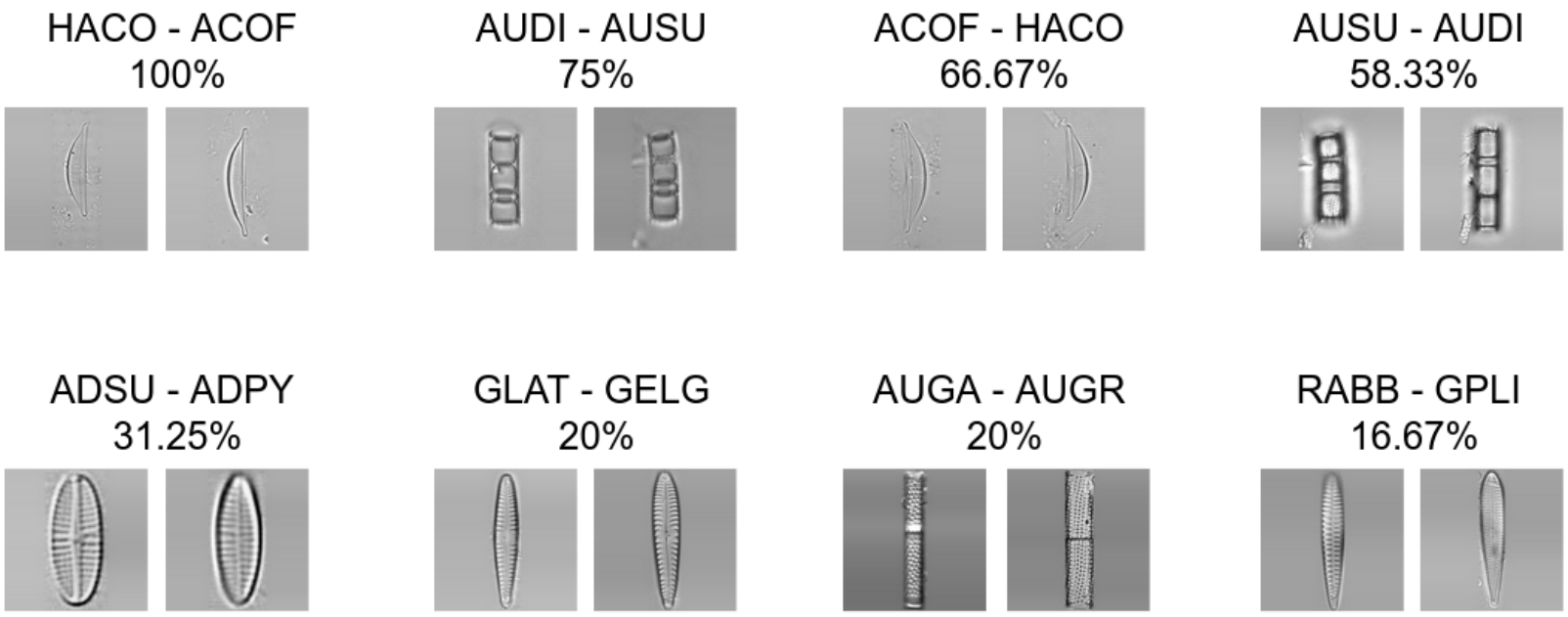


Figure 5.5: Examples of species misclassified by the classifier. For each pair of images in a row, the left image belongs to the ground truth species and the right image is from the predicted species. The scientific names for the letter codes are provided in Table A.1. The full list of misclassified taxa and the misclassification rate are provided in Table C.1.

5.3.3 Robustness analysis

Figure 5.6 shows the classification accuracy for the different values of the perturbation parameters (Gaussian blur, occlusion, random noise) in the sensitivity analysis. As the intensity of perturbation increased, the network's error rate increased, which is shown by the decrease in the classification accuracy. When the classification was performed at the species level of the hierarchy, the drop in accuracy was much more than the drop in accuracy observed when classifying at the genus level.

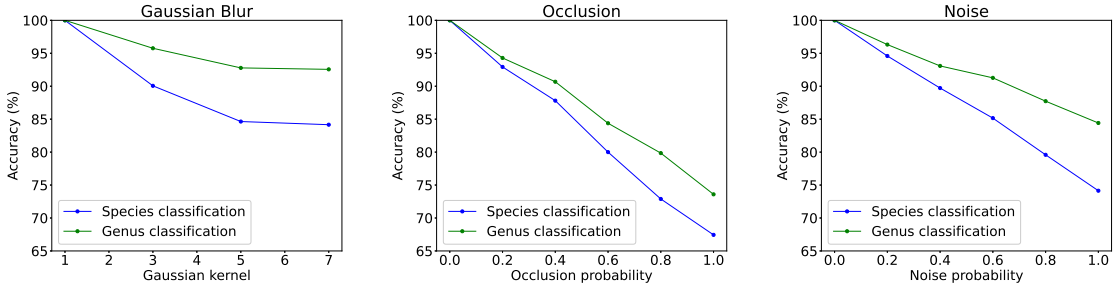
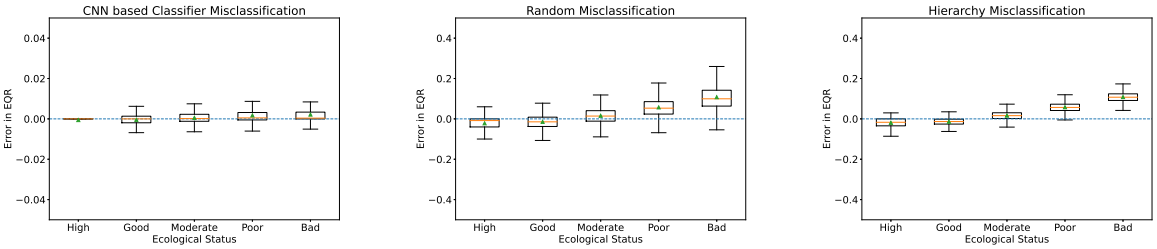


Figure 5.6: Analysis of robustness of the classification network in identifying input images subject to 3 different perturbations commonly observed in diatom images: image blurs (Gaussian Blur), occlusions (Occlusion), and random noise (Noise). The robustness was analysed when classifying at species level (blue) and genus level (green).

5.3.4 Misclassification analysis

Introducing misclassification changes the list of species and their abundances, which further modifies the BDI and EQR scores. In the following results, the error is described by the difference between the EQR scores respectively obtained before and after introducing misclassification. For the analyses, we considered misclassification rates from 10% to 100%, with successive steps of 10%. As shown in Section 5.3.2, the maximal misclassification range in the data range was 21.27%. Therefore, the following results focus on the case in which 20% of the individual species in the simulated inventories were misclassified. Figure 5.7 shows the box plots of the EQR error for the 5 ecological status classes according to the 3 predefined scenarios: based on the flat classifier's misclassification rates (Figure 5.7(a)), introduced randomly (Figure 5.7(b)) or introduced at the genus level of diatom's taxonomic hierarchy (Figure 5.7(c)). The corresponding normalized confusion matrices of the 5 ecological status classes are shown in Figure 5.8. The values in the matrices represent the number of images from the ground truth labels that were assigned to a particular status and were predicted as a different status when introducing misclassification at 20% rate on the simulated data.

Results for the other misclassification rates (from 10 to 100%) are shown in Appendix C (Figures C.1 to C.6).



(a) (b) (c)

Figure 5.7: Box plot of the EQR errors (difference between the EQR scores obtained before and after introducing misclassification) for the 5 ecological status at 20% misclassification rate (a) introduced using a CNN-based classifier's confusion matrix, (b) introduced randomly or (c) introduced at genus level of diatom's taxonomic hierarchy. The orange line in each box plot represents the median value and the green point is the mean error. The box represents the interquartile range (IQR), the whiskers represent the minimum and maximum range in the data.

CNN-based classifier misclassification

From the box plot in Figure 5.7(a), the errors in the EQR scores were low for all the ecological status. Correspondingly, the changes in site ecological status were also minimal. At least 98% of simulations resulted in no change in the ecological status of sites, whatever the ecological status (as shown by the diagonal values in Figure 5.8(a)). The change of status from "Good" to "Moderate" (or "Moderate" to "Good"), which is of particular interest to the WFD, was only 1%. Results for the other misclassification rates show that the uncertainty in the EQR score increased with increasing misclassification rate, but remained low ($< 10\%$) even at high misclassification rates and regardless of the ecological status (Figures C.1 and C.2).

Random misclassification

Figure 5.7(b) shows that the errors in the EQR scores were low for the "High" and "Good" ecological status and increased for the "Poor" and "Bad" ecological status. This can again be visualized in the confusion matrix (Figure 5.8), where the correct predictions in the "Bad" class were as low as 43%. In terms of ecological status allocation, 14% of the inventories from the "Good" ecological status were allocated to the "Moderate" ecological status, and conversely 10% of the inventories from the "Moderate" ecological status were allocated to the "Good" ecological status, due to the misclassifications. These rates of change in ecological

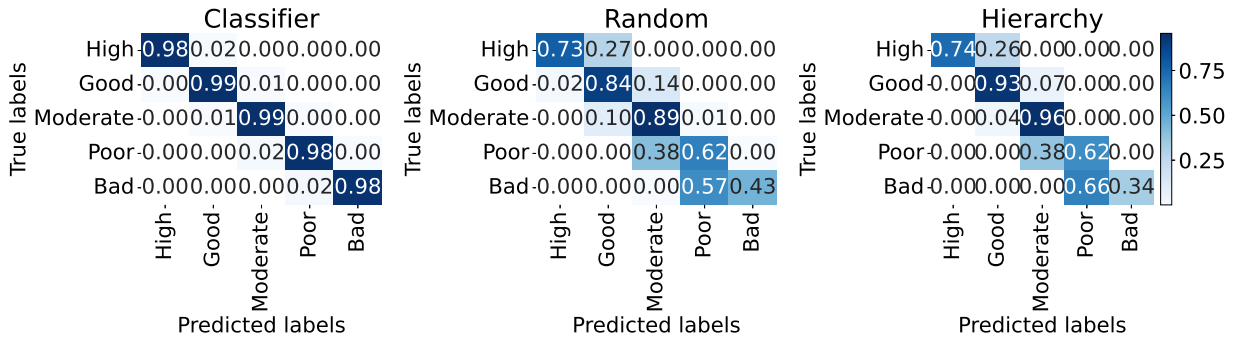


Figure 5.8: Normalized confusion matrices for the 5 ecological status (from "high" to "bad") at 20% misclassification rate a) introduced using a CNN-based classifier's confusion matrix, b) introduced randomly or c) introduced at genus level of diatom's taxonomic hierarchy. The X-axis is the predicted label and the Y-axis is the ground truth labels. The values in the matrices are the % of inventories that belong to a particular status in the ground truth labels and were predicted as another status when introducing misclassification.

status were not so balanced for the other between-class boundaries, with a clear overestimation of the ecological status of sites with the worst observed ecological status (poor: 38%; bad: 57%; Figure 5.8(b)) and a clear underestimation of the ecological status of sites with the highest ecological status (high: 27%). Results for the other misclassification rates show that the uncertainty in the EQR score also increased with increasing misclassification, and at a much higher rate than the CNN classifier. The maximum error observed was 94% at a misclassification rate of 100% (Figures C.3 and C.4).

Hierarchy-based misclassification

Similar to the random misclassification case, the errors in the EQR scores were low for the "High" and "Good" water classes and increased for the "Poor" and "Bad" classes, with a slight drop in the error variances (Figure 5.7(c)). This can again be visualized in the confusion matrix (Figure 5.8(c)), where the correct predictions in the "Bad" status was as low as 34%. In terms of ecological status allocation, 7% of the inventories from the "Good" status were allocated to the "Moderate" ecological status, and conversely 4% of the inventories from the "Moderate" ecological status were allocated to the "Good" ecological status due to the a 20% misclassification rate. As for the random misclassification, a clear overestimation of the ecological status of sites with the worst observed ecological status (poor: 38%; bad: 66%; Figure 5.8(b)) and a clear underestimation of the ecological status of sites with the highest ecological status (high: 26%) were observed. Results for the other misclassification rates show

that the uncertainty in the EQR score also increased with increasing misclassification but the variance in uncertainty was less than the random misclassification case. For misclassification rates above 50%, all the 'bad' water classes had an error rate of 100%. (Figures C.5 and C.6).

5.4 Discussion and Conclusion

In this chapter, we investigated the impact of misclassifications on BDI score and the related ecological status allocation, in the context of automatic diatom classification using deep learning at the scale of the Rhine-Meuse basin. We simulated thousands of synthetic diatom observations and conducted a study on the simulated data to analyse the differences in BDI scores for varying levels of misclassifications obtained from the network's confusion matrix. To account for the uncertainties in the network's robustness, we analysed two scenarios: when the taxa are randomly misclassified, and misclassification at higher taxonomic level of the diatom hierarchy. Results showed that the BDI is robust to a classifier's misclassifications. When errors were randomly introduced, the BDI uncertainty increased. Hierarchical classification showed promising results in minimizing the uncertainty of the BDI scores. While our results are promising, there are several noteworthy points to consider for a more comprehensive understanding of the implications: Firstly, it is essential to acknowledge that our analysis was conducted on a limited number of diatom classes. To obtain a more precise assessment of uncertainty, encompassing all 455 diatom classes would be necessary. The inclusion of a broader spectrum of diatom species in future studies could provide a more comprehensive view of the BDI score's sensitivity to misclassifications and its implications for ecological assessment. Furthermore, the extent of error propagation is also contingent on whether the misclassified species share a similar ecological profile. The results of the CA analysis indicate that we effectively address this aspect for the Rhin-Meuse region. From the analyses, it can also be observed that misclassification rates remain relatively consistent when incorporating "random" or only at "genus level" species replacements. In fact, the rates are slightly higher in the latter scenario, as demonstrated by the confusion matrices in Figure 5.8. This suggests that the differences in ecological preferences between diatom species within the same genus are not notably narrower compared to differences between species from distinct genera. As a prospective direction for future research, it becomes crucial to explore additional case studies, encompassing a broader range of scenarios, such as misclassified species with distinct ecological profiles, in various geographical basins or settings. Secondly, we should recognize that the performance of the classifier could further improve with an increased number of images. A larger and more diverse dataset would po-

tentially enable the model to better capture the subtle distinctions between diatom species, subsequently reducing the error rate. Therefore, ongoing efforts to expand the dataset and enhance the training process may yield more accurate results and lower BDI uncertainties. However, it is crucial to consider the potential trade-off when training the network with a more extensive set of diatom classes. As the number of classes increases, the inter-class similarity within the diatom hierarchy may also rise. This heightened similarity can lead to an elevated misclassification rate, consequently increasing the BDI uncertainty. In addition to the factors discussed above, it is important to address the quality of the diatom images used in our study. Factors such as image resolution, lighting conditions, focus, and image noise can influence the classifier's ability to accurately identify and classify diatom species. Images with low resolution or significant noise levels, for instance, may pose challenges to the classifier's ability to discern fine morphological details, potentially leading to misclassifications and increased BDI uncertainty.

Chapter 6

Hierarchy for Image Retrieval

6.1 Introduction

In recent years, the exponential growth of image data has made effective management of such data increasingly important. One way to manage this data is through Content-based Image Retrieval (CBIR), where a query image is used to retrieve images from the database that are visually similar to it [30]. The comparison is performed on the features extracted from the images, represented by a set of numerical descriptors or vectors. The search is performed on these descriptors using a similarity measure such as the Euclidean or cosine distance. The challenge is to extract robust feature representations, such that objects belonging to the same category are close by, despite the changes in view-point, background and illumination. Recent methods employing CNN for feature extraction have proven to be robust and effective over the classical hand-crafted feature extractors [187]. A commonly adopted approach is to extract feature vectors from the latent space of a trained CNN classifier. Deep classifiers use cross-entropy loss that learns feature representations to group together objects belonging to the same category, while separating them from the rest. Thus, given a query image, the retrieved data from the local neighbourhood in the latent space should ideally consist of images belonging to the same classification category.

Despite the ability of the CNNs to learn discriminative representations, it has been observed that sometimes, images unrelated to the query are retrieved [9, 136]. These images are typically visually similar to the query, but semantically very different. One way to address this is by incorporating domain knowledge while training, so that the network learns feature representations that are both visually and semantically relevant [9]. The semantic data is obtained from expert domain knowledge or lexical databases such as WordNet [125]. However, incorporating external domain knowledge during training can be challenging since it would require significant changes in the training procedure, and can be detrimental when

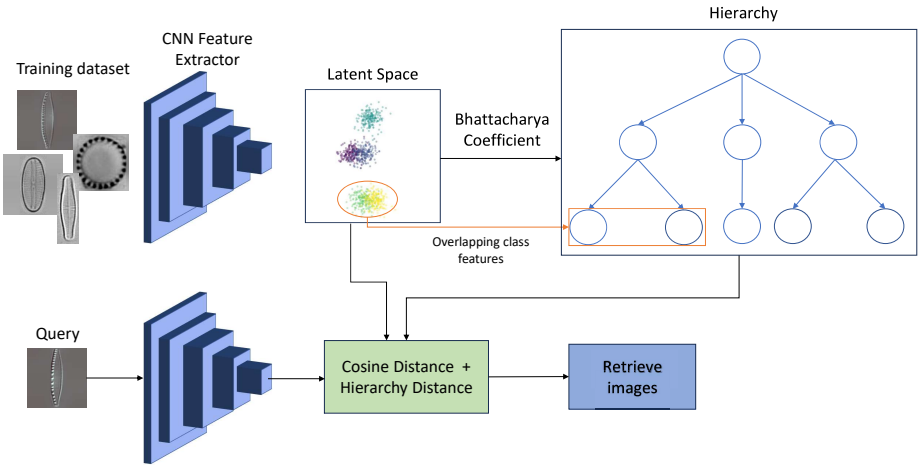


Figure 6.1: Illustration of our proposed method for image retrieval. Our approach uses representations learned by a classification network to identify overlapping classes. These classes are grouped together at different levels to create a hierarchy. When a query image is obtained, a combination of the cosine distance and a hierarchy distance is used to rank the images for retrieval.

there are inconsistencies between visual and semantic similarities [17]. Also, it assumes that the specialized domain knowledge is available, which can be hard to obtain or irrelevant in certain situations, such as, querying the images of people or vehicles. In this paper, we demonstrate that the visual hierarchies obtained from the representations learned by the classification network are rich in visual and semantic information, achieving superior retrieval performance. Contrary to other approaches, our method does not require specialized domain knowledge or modifications to the training procedure, making it easy to integrate into an off-the-shelf classification network.

Our method, illustrated in Figure 6.1, relies on two key aspects for the retrieval: construction of a hierarchy that is visually and semantically meaningful, and integration of hierarchy into the distance calculation metric for similarity search. Our method for hierarchy construction involves identifying the classes in the latent space whose feature representations overlap with one another. This is based on the assumption that the overlapping classes are those that are very similar to each other, both visually and semantically. The clusters of overlapping classes are then grouped together into the same class at a higher hierarchy level.

This process is repeated at different levels of the hierarchy until there are no overlaps in the feature representations, resulting in a hierarchical structure that captures both visual and semantic similarities between images. Once the hierarchy is obtained, we combine the cosine distance and a hierarchy-based distance to obtain a metric for ranking the images for retrieval.

Experimental results show that our method improves image retrieval performance for diatoms. To assess the generalizability of our approach, we evaluate on two commonly used benchmarks: CIFAR100 [93] and CUB-200-2011 [186]. Finally, we demonstrate the robustness of our method by evaluating on images subject to perturbations that mimic practical difficulties encountered during acquisition. The work described in this chapter is based on the publication in *ICVS 2023* [182]. The code is available in https://github.com/vaishwarya96/Hierarchy-image-retrieval.git.

6.2 Related Works

The role of visual attributes and semantics is crucial in describing images for identification or retrieval. Hierarchies are an intuitive way to express this, and several methods have leveraged it for image categorization and retrieval [163]. This is done by integrating hierarchy obtained from the biological taxonomy or from lexical databases to learn feature representations that capture semantic similarity between the categories. [18] uses WordNet to construct hierarchies and incorporates the domain knowledge obtained from it into a CNN classifier to improve the network's performance. [12] extends the traditional cross-entropy loss used in flat classification to work on hierarchies. To do so, they compute the conditional probability of the nodes at each layer of the hierarchy and calculate the cross-entropy loss at each layer. DeViSE [53] transfers semantic knowledge from text to visual object recognition. It involves pre-training a language model to learn word embeddings and re-training a deep visual model using these embeddings for image label prediction. [9] uses an embedding algorithm that maps images onto a hypersphere. The distances on this hypersphere represent similarities that are derived from the lowest common ancestor height in a given hierarchy tree. One limitation of these methods is their reliance on an existing domain knowledge, which can be difficult to obtain or irrelevant in some contexts.

A line of work relies on the visual features learnt by the network. [196] uses metric loss while training the network for image retrieval. While this reinforces the condition that the intra-class representations are clustered tightly, it does not impact the semantic relationship between the instances. [202] constructs a visual hierarchy from the distance values between the image features and uses it to improve classification. Our method is similar to [136],

which uses Ward clustering for constructing the hierarchy and uses it for re-ranking the images retrieved using a distance metric. A drawback of using Ward clustering is that they are sensitive to outliers and can sometimes group classes with low semantic similarities. In contrast, our approach groups classes only when their features overlap, which results in a more meaningful and semantically relevant hierarchy.

6.3 Method

This section explains the two main steps of our method: the construction of the hierarchy from the learned feature representations, and the calculation of the distance metric for ranking the images. The notations in this chapter follow the same as Chapter 4.

6.3.1 Hierarchy Construction

The hierarchy construction involves identifying the classes with overlapping distributions of feature vectors in x_{train}. Inter-class overlapping typically occurs when there are visual and semantic similarities in the class images. To construct the hierarchy, a bottom-up approach is used, where the overlapping classes in the latent space are merged in the higher hierarchy level. The overlapping classes are identified using the Bhattacharya Coefficient (BC), which is a statistical metric to measure the amount of overlap between two statistical populations.

The Bhattacharya distance between two classes C_i and C_j is given by:

$$D_B(C_i, C_j) = \frac{1}{8}(\mu_i - \mu_j)^T \mathbf{\Sigma}^{-1}(\mu_i - \mu_j) + \frac{1}{2}\ln\left(\frac{\det \mathbf{\Sigma}}{\sqrt{\det \boldsymbol{\Sigma_i} \det \boldsymbol{\Sigma_j}}}\right) \tag{6.1}$$

where $\mathbf{\Sigma} = \frac{\boldsymbol{\Sigma_i}+\boldsymbol{\Sigma_j}}{2}$. The term $(\mu_i - \mu_j)^T \mathbf{\Sigma}^{-1}(\mu_i - \mu_j)$ is the Mahalanobis Distance, which was used for uncertainty quantification in Chapter 4.

The BC is calculated as:

$$BC(C_i, C_j) = e^{-D_B(C_i, C_j)} \tag{6.2}$$

The value of BC ranges from 0 to 1 inclusive, where a value of 0 signifies no overlap and 1 means full overlap. If the value of the BC between two classes is greater than or equal to a specified threshold t, then they are considered to be overlapping. Every cluster of overlapping classes are merged together at the higher level of hierarchy to create a new class. On the other hand, if the BC is less than t, then these classes are considered to be distinct and are kept separate in the higher hierarchy level. The t value is a hyperparameter that depends on the dataset. The process of BC estimation is repeated at each level on the

new set of classes, and the overlapping classes are merged to create a hierarchy structure. This is performed until there are no overlaps between the classes, or all the classes have been merged. It should be noted that at the higher hierarchy levels, some classes include feature vectors from multiple small classes that were merged. At every level, the $\boldsymbol{\mu}$ and $\boldsymbol{\Sigma}$ values are recalculated from the feature vectors of the merged classes.

6.3.2 Distance Calculation

After constructing the hierarchy, we use it to perform the similarity search for image retrieval. The distances are calculated between the query feature x_q and the image features x_i in the database. Our final distance metric for ranking the images is a weighted sum of two distance values: cosine distance and a hierarchy based distance.

Cosine distance: The cosine distance is a similarity measure between two feature vectors and is popularly used in image retrieval tasks. The cosine distance between x_q and x_i is given by:

$$D_C(x_q, x_i) = 1 - \frac{x_q \cdot x_i}{\|x_q\|\|x_i\|} \tag{6.3}$$

where $\cdot$ is the dot product and $\|\ \|$ is the L2 norm.

Hierarchical distance: The hierarchy is a tree structure $\mathcal{H} = (V, E)$ where V represents the nodes and E, the edges. $\mathcal{H}$ consists of K leaf nodes. We want to identify the position of any database image in the hierarchy, which means finding the leaf closest to each image. To identify the leaf nodes, we calculate the MD between the feature vectors x_i and the feature vectors x_{train} corresponding to each leaf node. The MD between x_i and node c, is given as:

$$MD_c(x_i) = \sqrt{(x_i - \mu_c)^T {\boldsymbol{\Sigma_c}}^{-1} (x_i - \mu_c)} \tag{6.4}$$

x_i is assigned to the leaf node n_i corresponding to the smallest MD value.

Similarly, the leaf node n_q corresponding to x_q is identified. The distance calculation uses the height of the lowest common ancestor (LCA), where the LCA of two nodes is the parent node that is an ancestor of both nodes and does not have any child that is also an ancestor of both nodes. The hierarchical distance [9] between x_q and x_i is calculated as:

$$D_H(x_q, x_i) = \frac{\text{height}(\text{LCA}(\text{n}_\text{i}, \text{n}_\text{q}))}{\text{height}(\mathcal{H})} \tag{6.5}$$

The final distance is given as:

$$D = D_C(x_q, x_i) + \alpha D_H(x_q, x_i) \tag{6.6}$$

where α is a hyperparameter.

The images are sorted in ascending order of D to rank the images based on similarity and retrieve them.

6.3.3 Robustness Analysis

We consider the following scenarios: (i) Blurring – The images are subject to Gaussian blurring, simulating the effect of an out-of-focus camera or a low-quality image (ii) Saturation change – The saturation values of the original images are modified to simulate variations in colour intensity. (iii) Occlusions – Random occlusions are introduced in different parts of the images, simulating objects or regions being partially or completely blocked.

6.4 Experiments

This section provides the baselines, experimental details and the metrics used for evaluating our method.

6.4.1 Baselines

We compare our method against the following approaches: (i) Image retrieval using Euclidean Distance, (ii) Image retrieval using Cosine Distance, (iii) Our method using hierarchical distance only, (iv) Semantic Embeddings [9], (v) Our method applied on the features extracted from semantic embeddings in (iv), (vi) Distance calculation using the method in Sec. 6.3.2 on hierarchy constructed using Ward clustering [136], (vii) Distance calculation on semantic hierarchy obtained using domain knowledge.

6.4.2 Experimental Setup

This section lists the datasets that were used for evaluating the method and the training details. We perform our experiments on three datasets with fine-grained visual features:

1. **CUB-200-2011 [186].** This is a widely used dataset for image retrieval, consisting of 200 classes of birds and a total of 11,788 images. The train data consists of 5,994 images and the test data consists of 5,794 images. 80% of the train dataset was used for training and 20% for validation. The images were resized to 224×224, and random horizontal flipping and rotation ($\pm 30°$) were used for data augmentation. Following [9], we used ResNet-50 [71] for training with an SGD optimizer. The network was trained for 100 epochs with a batch size of 12 on GeForce RTX 3090 with 24 Gb of RAM. The value of $t = 0.30$ and $\alpha = 3$, obtained using hyperparameter search.

2. **CIFAR100 [93].** It consists of 60,000 images, of size 32×32 belonging to 100 classes of wide range of objects. The training dataset consists of 50,000 images and the remaining are test data. 80% of the train dataset was used for training and 20% for validation. Following [9], we used ResNet-110 [71] for training with an SGD optimizer. The network was trained for 400 epochs with a batch size of 24 on GeForce RTX 3090 with 24 Gb of RAM. The value of $t = 0.20$ and $\alpha = 5$, obtained using hyperparameter search.

3. **Diatom Dataset.** This is the dataset described in Sec. 4.1.3, which is a subset of the Atlas dataset. 80% of the train dataset was used for training and 20% for validation. The network was trained using EfficientNet [169]. An SGD optimizer with a learning rate of 0.0002 was used. The images were resized to 256×256. The network was trained for 100 epochs with a batch size of 12 on GeForce RTX 3090 with 24 Gb of RAM. The value of $t = 0.25$ and $\alpha = 3$, obtained using hyperparameter search.

The semantic hierarchy of CUB-200-2011 and CIFAR100 for the baseline experiments were obtained from [9]. The diatom hierarchy was obtained from the taxonomy of the species, consisting of two levels: 'genus' and 'species'. The same model architecture was used for comparing all the baselines.

Setup for Robustness Analysis: The analysis is performed on the CUB-200-2011 dataset with the following settings. (i) A Gaussian Blur with kernel size (11,11) was applied to all the images, (ii) The saturation of images were adjusted with randomly chosen values in the range [0,1,0.5,1,2,5] and (iii) A random crop of dimension (70,70) was replaced by

black patch in every image. The networks were retrained on the modified images, with the same training parameters.

6.4.3 Evaluation Metrics

We use MAP@k values for evaluating the image retrieval performance. MAP@k measures the average precision (AP) of the top k retrieved images, and then takes the mean across all queries. A higher value indicates better performance.

6.5 Results

Figure 6.2 shows the plots of mAP@k values for different values of k for the three datasets. The results suggest that combining the visual and semantic cues improves the retrieval. Overall, our method achieves the best performance over the other baselines.

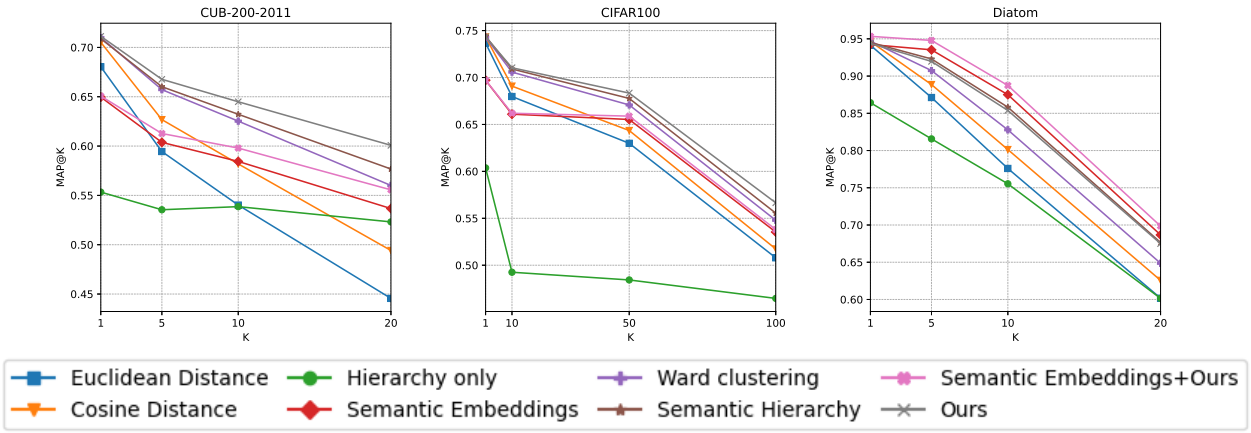


Figure 6.2: Retrieval performance on CUB-200-2011, CIFAR100 and Diatom dataset. Our method achieves state-of-the-art performance. The results suggest that incorporating hierarchy into content-based image retrieval is effective in improving the retrieval performance.

Effect of distances: The cosine distance retrieves images that are visually similar to the query. Whereas, the hierarchical distance takes into account the semantic relationships between classes. Both these aspects are important when retrieving the images, and removing one of them deteriorates the performance. There is a significant drop in the MAP values in CIFAR100 when using only the hierarchical distance. One explanation for this is that CIFAR100 contains classes that are visually diverse, and hence, visual cues are more important over the semantic ones here. This could also explain the marginal improvement in the

metrics when combining both the visual and semantic attributes than when using only the visual information for CIFAR100 over the other datasets.

Effect of hierarchy: In the hierarchy-based methods, our approach outperforms the baselines on CUB-200-2011 and CIFAR100. Semantic embedding [9], which relies on domain knowledge for feature learning, has a better performance on the diatom dataset. For the diatoms, the biological taxonomy is constructed based on the morphological features, and thus, the visual and semantic features are strongly correlated.

On the other two datasets, the drop in performance of semantic embeddings could be attributed to the errors introduced due to disagreements between the visual and semantic relationships while training [17]. Our approach of hierarchy construction merges the classes only when there is a correlation between visual and semantic attributes, and thus it is robust against these errors. The Ward hierarchy approach may sometimes group classes that are semantically unrelated, resulting in less effective performance. Semantic hierarchy uses domain knowledge, which again faces the problem of errors due to uncorrelated visual and semantic relations. To conclude, when the semantic attributes from the hierarchy correlates with the visual features, the retrieval performance is better.

Qualitative Analysis: Figures 6.3, 6.4 and 6.5 show some images retrieved using cosine distance and our method on the CIFAR100, CUB-200-2011 and the diatom dataset respectively. While cosine distance retrieves images that are visually similar, some of them are semantically very different. Whereas, our method retrieves images that are visually and semantically meaningful.

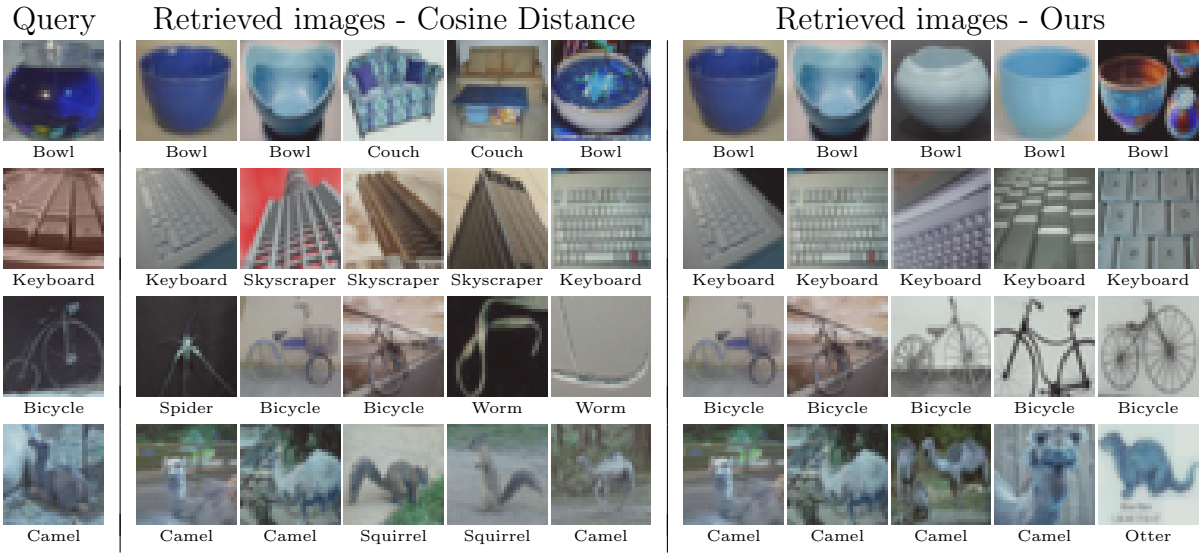


Figure 6.3: Examples of top-5 images retrieved using cosine distance and our method on CIFAR100. Our method incorporates hierarchy while ranking image similarity, with retrieves images that are more visually and semantically relevant.

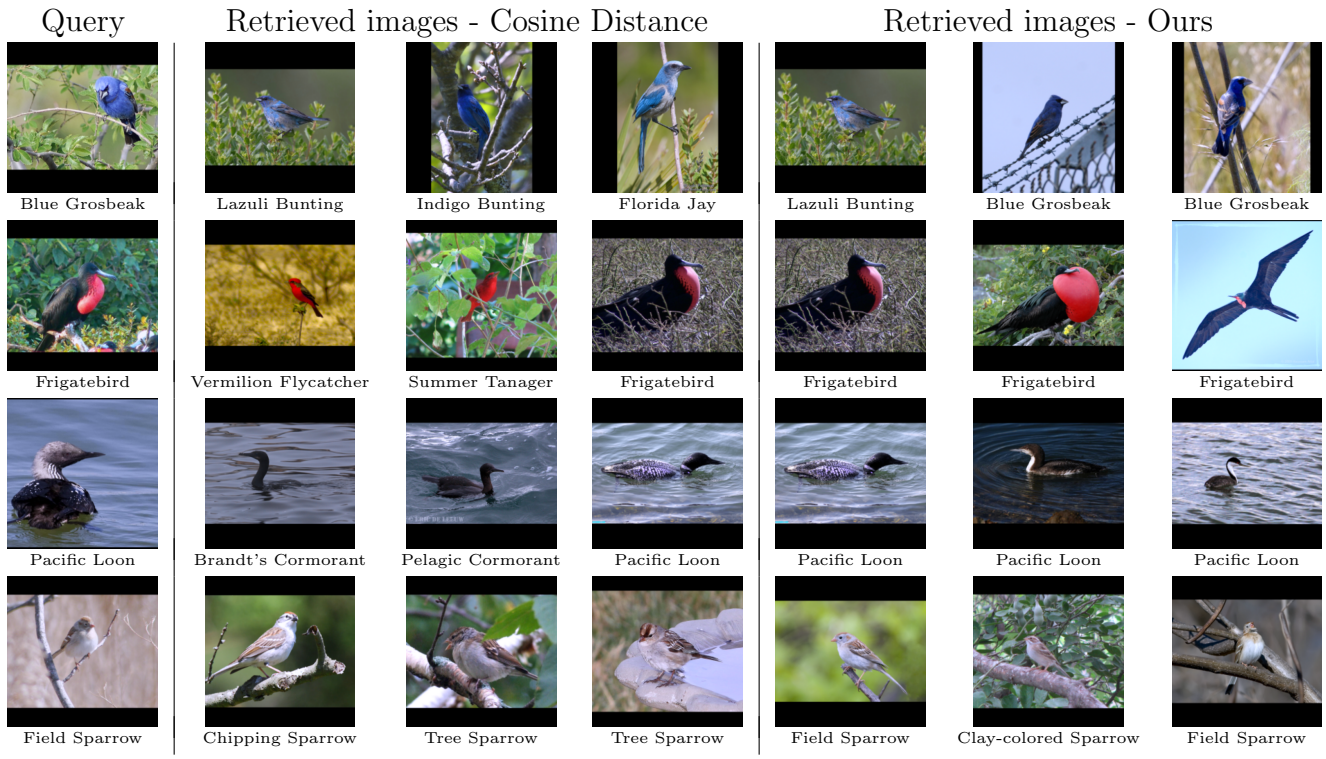


Figure 6.4: Examples of top-3 images retrieved using cosine distance and our method on the CUB-200-2011 dataset. Our method incorporates hierarchy while ranking image similarity, with retrieves images that are more visually and semantically relevant.

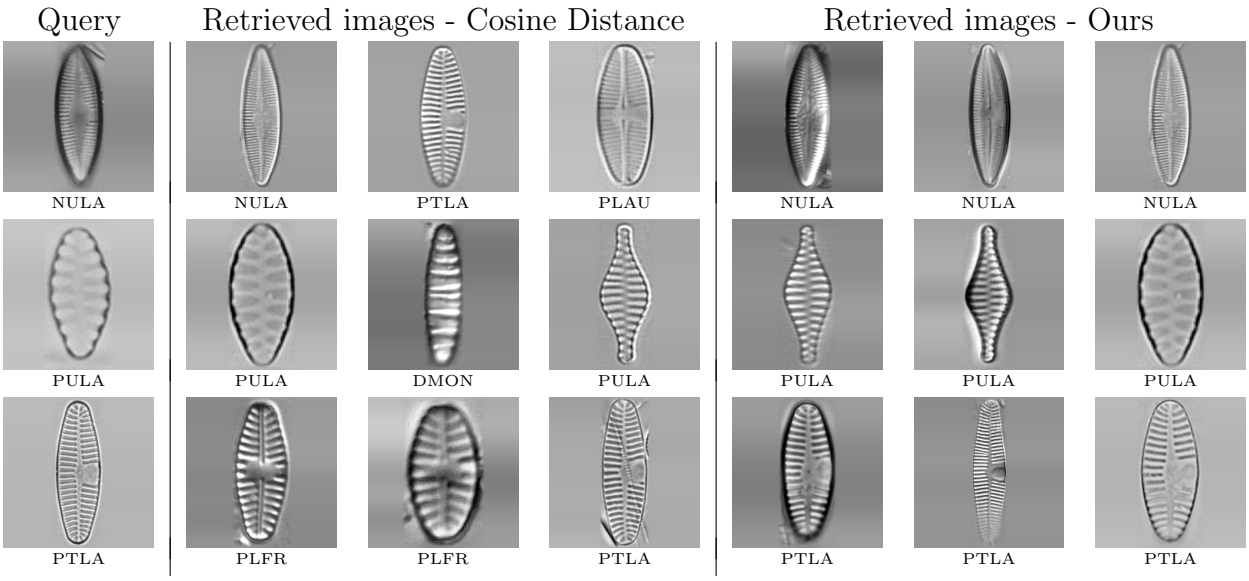


Figure 6.5: Examples of top-3 images retrieved using cosine distance and our method on Diatom dataset. Our method incorporates hierarchy while ranking image similarity, with retrieves images that are more visually and semantically relevant.

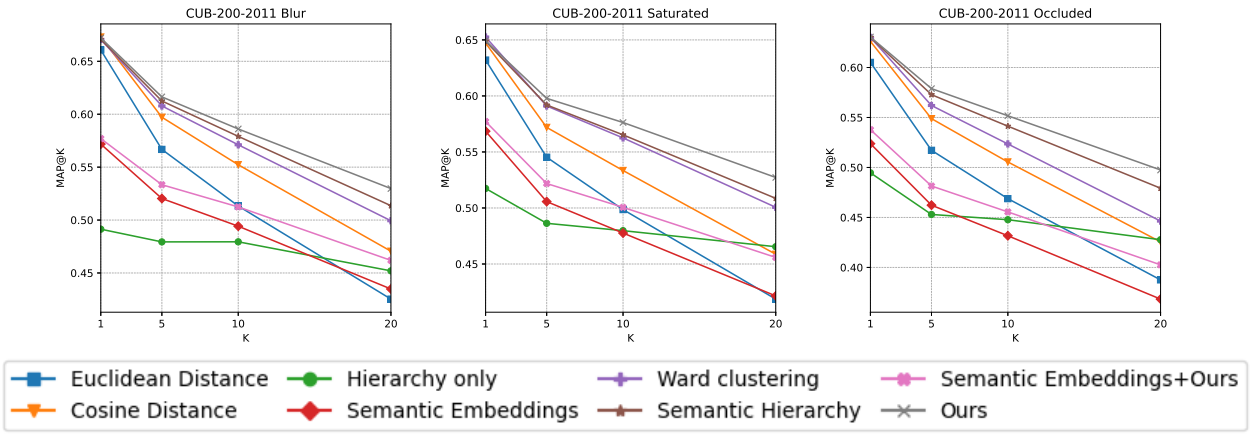


Figure 6.6: Robustness analysis on the CUB-200-2011 dataset. Blurring, saturation change and occlusion were applied to test the impact on the retrieval. Results show that our method still achieves the best performance.

Robustness Analysis: Figure 6.6 shows the MAP@K plots for the robustness analysis on CUB-200-2011. Compared to the plot in Figure 6.2, one can observe a drop in the metrics. However, our method still achieves the best performance over the other baselines, indicating its robustness even in challenging conditions.

Classification Results: The classification accuracy obtained for CUB-200-2011, CIFAR100 and Diatoms are 74.51%, 77.53% and 94.38% respectively. The accuracy obtained using the semantic embeddings training are 72.76%, 76.48% and 95.81% respectively. The slight drop in the CUB-200-2011 and CIFAR100 accuracy for semantic embeddings could be attributed to the errors introduced due to disagreements between the visual and semantic relationships while training, as observed by [17]. Whereas, the accuracy is marginally higher for the diatom dataset. The biological taxonomy is constructed based on the morphological features, and thus, the visual and semantic features are strongly correlated. Thus, semantic embeddings are beneficial here.

Failure Cases: Figure 6.7 shows some failure cases while retrieving diatom images. The retrieved images are visually very similar to the query image. The fine-grained nature of some diatom classes makes it challenging for the algorithm to accurately retrieve the images belonging to the same class.

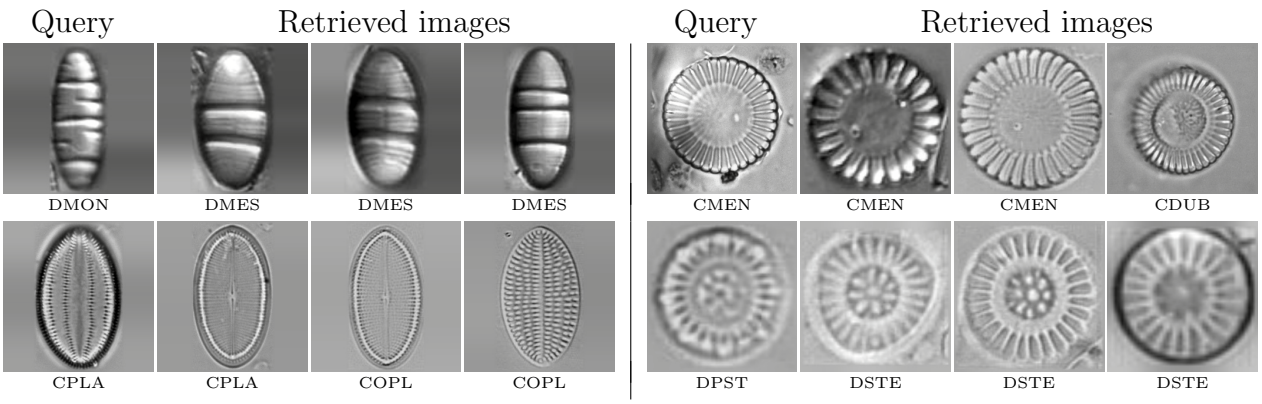


Figure 6.7: Failure cases of our approach on the diatom dataset.

6.6 Conclusion

In this chapter, we presented a method to tackle the problem of semantically dissimilar images retrieved in CBIR. By leveraging the learned feature representations from a CNN classifier, we construct a meaningful hierarchy that captures both visual and semantic information. This hierarchy is then integrated into the distance metric for similarity search, resulting in superior image retrieval performance, as demonstrated on fine-grained visual datasets. This tool can be valuable for biologists aiming to swiftly locate analogous diatom images, serving the purpose of comparison and identification. Through the utilization of an image retrieval mechanism, scientists can submit an image of a diatom they wish to identify, prompting the system to fetch similar diatom images from an extensive dataset based on their visual attributes. This expedites the process of narrowing down potential matches and furnishes an initial point of inquiry.

Chapter 7

Conclusion and Future Work

This thesis dealt with the task of automating the diatom identification process, aiming to streamline and enhance the accuracy of water quality assessments. Diatoms, as essential components of aquatic ecosystems, offer valuable information on environmental conditions. The conventional manual identification process is laborious, time-consuming, and subject to human biases. Through the development of an end-to-end pipeline powered by deep learning techniques, this work aims to alleviate several of the challenges faced during manual identification.

In addition to its focus on automating the identification process, a contribution of the thesis is that it addresses some prevalent challenges encountered during automation. One frequently encountered issue when training deep learning models pertains to the scarcity of labelled datasets. To address this, a method was introduced to generate synthetic microscopy images. The synthesized images were then employed to train object detection networks, enhancing their performance. To ensure the robustness of the proposed approach, a comprehensive sensitivity analysis was conducted on the different parameters utilized for generating the synthetic data. This assessment shed light on the intricate relationship between these parameters and the resulting model performance. Moreover, the practicality of the automated detector, particularly the instance segmentation network, was demonstrated through its application in a specific use-case on environmental assessment. This involved an exploration of how various toxics impact the morphology of diatoms. The masks obtained from instance segmentation were used to extract the various statistical and textural descriptors, which were further used to gain insights into the diatom morphology.

Furthermore, the thesis confronts the issues arising from both inter-species similarities and intra-specie variations, which can hinder accurate classification. A self-supervised representation learning method was introduced that involves grouping visually similar images of a species together through clustering, while simultaneously using metric learning to create

distinctions between representations of distinct species. The developed method was further applied to measure the level of uncertainty in predictions derived from deep classifiers and to identify samples that fall outside the known distribution (Out-of-Distribution or OOD). The uncertainty estimation was performed by gauging the proximity of a test sample to the distribution of training data, specifically utilizing the Mahalanobis distance. Through this approach, well-calibrated prediction probabilities were obtained, enabling us to not only quantify uncertainty but also effectively identify instances of OOD data.

The practical utility of the developed tool was showcased in a real-world scenario—specifically, the calculation of the Biological Diatom Index (BDI) for water quality assessment. To understand the influence of misclassifications on BDI accuracy, an extensive analysis was conducted using synthetically generated abundance data. This analysis aimed to delve into the propagation of errors introduced at different phases of the pipeline on the ultimate BDI score.

Finally, an image retrieval system was also devised. This system allows researchers to input images of diatoms they wish to identify, streamlining the process by swiftly narrowing down potential matches and offering a preliminary direction for in-depth exploration. The methodology employed the creation of a hierarchical structure that considers both semantic and visual similarities among images. This hierarchical approach was integrated into conventional image retrieval methods, leading to enhancements in retrieval performance.

While the current study marks a step towards automating diatom identification, there remain several avenues for future exploration and refinement.

- **Real-world Validation:** While the presented tool exhibits considerable promise, its efficacy in diverse real-world scenarios is a fertile ground for investigation. The detection and classification tools have been developed with diatom images acquired from the Rhin-Meuse basin. Validating the tool's performance on other river basins and across a range of water quality conditions would be invaluable in assessing its robustness and applicability.

- **Interpretability and Explainability:** As deep learning models often function as "black boxes", unravelling the rationale behind their decisions remains a pertinent concern. Developing techniques to elucidate the features that contribute to classification outcomes could enhance the tool's interpretability, thereby fostering trust and facilitating its adoption.

- **Classification Improvement:** In terms of further improving the classification performance, a key area of focus should be on advancing the robustness of classification,

especially when confronted with imbalanced datasets. When implementing diatom classification in real-time systems, the dataset continually evolves as taxonomy changes or additional data is incorporated to enhance network performance. As new data flows in, the existing data imbalance may further escalate, with certain species dominating in terms of the number of available training images compared to the rarer ones. To address these challenges, various methods have been proposed in the literature, such as class re-balancing and resampling [197, 203]. Additionally, the development of active learning algorithms can greatly assist in the data annotation process during dataset curation. Active learning algorithms actively select informative samples from the unlabelled data for annotation, reducing the overall labelling effort. By iteratively choosing the most informative instances based on certain criteria, such as uncertainty estimation or representation diversity, active learning algorithms maximize annotation efficiency and improve classification accuracy [150, 162]. Another important consideration is to leverage prior knowledge when identifying an individual, especially when other individuals have been identified with a high degree of confidence. The co-occurrence information can play a crucial role in identifying individuals that might be challenging to identify when examined in isolation.

- **Trait-based Classification:** In Chapter 4, the classification was performed based on the taxonomy of the diatoms. One could also classify diatoms based on the morphological traits exhibited, such as the diatom size, shape, deformation and biovolume as potential early indicators of anthropogenic stresses [123]. In Chapter 3.5, we developed a method to extract various morphological parameters from the segmentation masks of the diatoms. This tool can be extended to analyse the diatom functional diversity, and to develop reliable biomonitoring indices [100, 132]. Alternatively, diatom taxonomic identification could be enhanced by considering their traits. Trait-based strategies offer an additional benefit by circumventing potential constraints associated with taxonomy when developing biomonitoring tools. This is particularly valuable as the taxonomic identification of organisms might prove unnecessary in certain contexts.

- **Dealing with Noisy Data:** The annotation of training data is typically performed by several human annotators. This can introduce bias and noise in the ground truth labels. The development of robust training methods to deal with noisy labels is a potential future direction.

7.1 Acknowledgement

The PhD scholarship for AV was funded by ANR, France (ANR-20-THIA-0010) and Région Grand-Est, France. Additional financial support was provided by CNRS, France (ZAM LTSER Moselle) and Horizon Europe (iMagine – Grant agreement ID: 101058625).

Bibliography

[1] rolabelimg. https://github.com/cgvict/roLabelImg.

[2] Rotate-yolov5. https://github.com/XinzeLee/RotateObjectDetection.

[3] Labelbox. https://labelbox.com, 2023.

[4] Moloud Abdar, Farhad Pourpanah, Sadiq Hussain, Dana Rezazadegan, Li Liu, Mohammad Ghavamzadeh, Paul Fieguth, Xiaochun Cao, Abbas Khosravi, U Rajendra Acharya, et al. A review of uncertainty quantification in deep learning: Techniques, applications and challenges. *Information Fusion*, 76:243–297, 2021.

[5] Benjamin Alric, Olivier Dezerald, Albin Meyer, Elise Billoir, Romain Coulaud, Floriane Larras, Cedric P Mondy, and Philippe Usseglio-Polatera. How diatom-, invertebrate- and fish-based diagnostic tools can support the ecological assessment of rivers in a multi-pressure context: Temporal trends over the past two decades in france. *Science of The Total Environment*, 762:143915, 2021.

[6] Vijay Badrinarayanan, Alex Kendall, and Roberto Cipolla. Segnet: A deep convolutional encoder-decoder architecture for image segmentation. *IEEE transactions on pattern analysis and machine intelligence*, 39:2481–2495, 2017.

[7] Abhishek Balasubramaniam and Sudeep Pasricha. Object detection in autonomous vehicles: Status and open challenges. *arXiv preprint arXiv:2201.07706*, 2022.

[8] Nitpreet Bamra, Vikram Voleti, Alexander Wong, and Jason Deglint. Towards generating large synthetic phytoplankton datasets for efficient monitoring of harmful algal blooms. *arXiv preprint arXiv:2208.02332*, 2022.

[9] Björn Barz and Joachim Denzler. Hierarchy-based image embeddings for semantic image retrieval. In *2019 IEEE winter conference on applications of computer vision (WACV)*, pages 638–647. IEEE, 2019.

[10] Herbert Bay, Tinne Tuytelaars, and Luc Van Gool. Surf: Speeded up robust features. *Lecture notes in computer science*, 3951:404–417, 2006.

[11] Anne-Sophie Benoiston, Federico M Ibarbalz, Lucie Bittner, Lionel Guidi, Oliver Jahn, Stephanie Dutkiewicz, and Chris Bowler. The evolution of diatoms and their biogeochemical functions. *Philosophical Transactions of the Royal Society B: Biological Sciences*, 372:20160397, 2017.

[12] Luca Bertinetto, Romain Mueller, Konstantinos Tertikas, Sina Samangooei, and Nicholas A Lord. Making better mistakes: Leveraging class hierarchies with deep networks. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 12506–12515, 2020.

[13] Maurice-Yves Bey and Luc Ector. Atlas des diatomées des cours d'eau de la région rhône-alpes. tome 1. Centriques, Monoraphidées. tome 2. Araphidées, Brachyraphidées. tome 3. Naviculacées: Naviculoidées. tome 4. Naviculacées: Naviculoidées. tome 5. Naviculacées: Cymbelloidées, Gomphonematoidées. tome 6. Bacillariacées, Rhopalodiacées, Surirellacées. pages 1–1182. Direction Régionale de l'Environnement, de l'Aménagement et du Logement Rhône-Alpes, ISBN:978-2-11-129817-0, 2013.

[14] Charles Blundell, Julien Cornebise, Koray Kavukcuoglu, and Daan Wierstra. Weight uncertainty in neural network. In *International conference on machine learning*, pages 1613–1622. PMLR, 2015.

[15] Vinicius R Pereira Borges, Bernd Hamann, Thais G Silva, Armando AH Vieira, and Maria Cristina F Oliveira. A highly accurate level set approach for segmenting green microalgae images. In *2015 28th SIBGRAPI Conference on Graphics, Patterns and Images*, pages 87–94. IEEE, 2015.

[16] Roberto Brunelli. Template matching techniques in computer vision: theory and practice. pages 1–348, 2009.

[17] Clemens-Alexander Brust and Joachim Denzler. Not just a matter of semantics: The relationship between visual and semantic similarity. In *41st DAGM GCPR 2019, Dortmund, Germany, September 10–13, 2019, Proceedings 41*, pages 414–427. Springer, 2019.

[18] Clemens-Alexander Brust and Joachim Denzler. Integrating domain knowledge: using hierarchies to improve deep classifiers. In *5th Asian Conference on Pattern Recognition, Revised Selected Papers, Part I*, pages 3–16. Springer, 2020.

[19] Gloria Bueno, Oscar Deniz, Anibal Pedraza, Jesús Ruiz-Santaquiteria, Jesús Salido, Gabriel Cristóbal, María Borrego-Ramos, and Saúl Blanco. Automated diatom classification (Part A): handcrafted feature approaches. *Applied Sciences*, 7:753, 2017.

[20] Andrea Buffagni, Stefania Erba, Marcello Cazzola, Emanuele Barca, and Carlo Belfiore. The ratio of lentic to lotic habitat features strongly affects macroinvertebrate metrics used in southern europe for ecological status classification. *Ecological Indicators*, 117:106563, 2020.

[21] Yannick Le Cacheux, Herve Le Borgne, and Michel Crucianu. Modeling inter and intra-class relations in the triplet loss for zero-shot learning. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 10333–10342, 2019.

[22] Simon Carbonnelle and Christophe De Vleeschouwer. Intraclass clustering: An implicit learning ability that regularizes DNNs. In *International Conference on Learning Representations*, 2020.

[23] Mathilde Caron, Piotr Bojanowski, Armand Joulin, and Matthijs Douze. Deep clustering for unsupervised learning of visual features. In *Proceedings of the European conference on computer vision*, pages 132–149, 2018.

[24] A Chandesris and H Pella. Appui scientifique à la mise en oeuvre de la directive cadre sur l'eau. constitution d'une base d'information spatialisée barrages, retenues et plans d'eau au niveau national en vue d'évaluer les modifications hydro-morphologiques. 2006.

[25] Lin Chang, Ruchen Wang, Haiyong Zheng, Jialun Dai, and Bing Zheng. Phytoplankton feature extraction from microscopic images based on surf-pca. In *OCEANS 2016-Shanghai*, pages 1–4. IEEE, 2016.

[26] Liang-Chieh Chen, George Papandreou, Iasonas Kokkinos, Kevin Murphy, and Alan L Yuille. Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs. *IEEE transactions on pattern analysis and machine intelligence*, 40:834–848, 2017.

[27] Liang-Chieh Chen, George Papandreou, Florian Schroff, and Hartwig Adam. Rethinking atrous convolution for semantic image segmentation. *arXiv preprint arXiv:1706.05587*, 2017.

[28] Liang-Chieh Chen, Yukun Zhu, George Papandreou, Florian Schroff, and Hartwig Adam. Encoder-decoder with atrous separable convolution for semantic image segmentation. In *Proceedings of the European conference on computer vision*, pages 801–818. Springer, 2018.

[29] Tianqi Chen, Emily Fox, and Carlos Guestrin. Stochastic gradient hamiltonian monte carlo. In *International conference on machine learning*, pages 1683–1691. PMLR, 2014.

[30] Wei Chen, Yu Liu, Weiping Wang, Erwin M Bakker, Theodoros Georgiou, Paul Fieguth, Li Liu, and Michael S Lew. Deep learning for instance retrieval: A survey. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 45:7270–7292, 2023.

[31] Weihua Chen, Xiaotang Chen, Jianguo Zhang, and Kaiqi Huang. Beyond triplet loss: a deep quadruplet network for person re-identification. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 403–412, 2017.

[32] Ralph T Clarke. Estimating confidence of european wfd ecological status class and wiser bioassessment uncertainty guidance software (wiserbugs). *Hydrobiologia*, 704:39–56, 2013.

[33] Ralph T Clarke and Daniel Hering. Errors and uncertainty in bioassessment methods—major results and conclusions from the star project and their application using starbugs. *The Ecological Status of European Rivers: Evaluation and Intercalibration of Assessment Methods*, pages 433–439, 2006.

[34] COCO. Coco - evaluate - metrics. http://cocodataset.org/#detection-eval, 2019. Accessed: 2020-05-05.

[35] Michel Coste, Sébastien Boutry, Juliette Tison-Rosebery, and François Delmas. Improvements of the Biological Diatom Index (BDI): Description and efficiency of the new version (BDI-2006). *Ecological indicators*, 9:621–650, 2009.

[36] Jialun Dai, Ruchen Wang, Haiyong Zheng, Guangrong Ji, and Xiaoyan Qiao. Zooplanktonet: Deep convolutional network for zooplankton classification. In *OCEANS 2016-Shanghai*, pages 1–6. IEEE, 2016.

[37] Navneet Dalal and Bill Triggs. Histograms of oriented gradients for human detection. In *2005 IEEE computer society conference on computer vision and pattern recognition (CVPR'05)*, volume 1, pages 886–893. IEEE, 2005.

[38] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*, pages 248–255. IEEE, 2009.

[39] Jean-Pierre Descy and Michel Coste. Utilisation des diatomées benthiques pour l'évaluation de la qualité des eaux courantes. *Contrat CEE B-71-23. Rapport final. Cemagref*, 1990.

[40] Or Dinari and Oren Freifeld. Variational-and metric-based deep latent space for out-of-distribution detection. In *The 38th Conference on Uncertainty in Artificial Intelligence*, 2022.

[41] Ellen M Ditria, Sebastian Lopez-Marcano, Michael Sievers, Eric L Jinks, Christopher J Brown, and Rod M Connolly. Automating the analysis of fish abundance using object detection: optimizing animal ecology with deep learning. *Frontiers in Marine Science*, 7:1–9, 2020.

[42] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929*, 2020.

[43] Karsten Mikael Dromph, Susana Agusti, Alberto Basset, Javier Franco, Peter Henriksen, John Icely, Sirpa Lehtinen, Snejana Moncheva, Marta Revilla, Leonilde Roselli, et al. Sources of uncertainty in assessment of marine phytoplankton communities. *Hydrobiologia*, 704:253–264, 2013.

[44] Hans Du Buf, Micha Bayer, Stephen Droop, Ritchie Head, Steve Juggins, Stefan Fischer, Horst Bunke, Michael Wilkinson, Jos Roerdink, José Pech-Pacheco, Gabriel Cristóbal, Hamid R. Shahbazkia, and Adrian Ciobanu. Diatom identification: a double challenge called ADIAC. In *Proceedings 10th International Conference on Image Analysis and Processing*, pages 734–739. IEEE, 1999.

[45] Shiv Ram Dubey, Satish Kumar Singh, and Bidyut Baran Chaudhuri. Activation functions in deep learning: A comprehensive survey and benchmark. *Neurocomputing*, 503:92–108, 2022.

[46] Debidatta Dwibedi, Ishan Misra, and Martial Hebert. Cut, paste and learn: Surprisingly easy synthesis for instance detection. In *Proceedings of the IEEE international conference on computer vision*, pages 1301–1310, 2017.

[47] Luc Ector and Daša Hlúbiková. Atlas des diatomées des alpesmaritimes et de la région provence-alpes-côte d'azur. *Belvaux: Centre de Recherche Public–Gabriel Lippmann*, 2010.

[48] Luc Ector, Carlos E Wetzel, MH Novais, and Didier Guillard. Atlas des diatomées des rivières des pays de la loire et de la bretagne. *DREAL Pays de la Loire, Nantes*, 649, 2015.

[49] Yan Em, Feng Gag, Yihang Lou, Shiqi Wang, Tiejun Huang, and Ling-Yu Duan. Incorporating intra-class variance to fine-grained visual recognition. In *2017 IEEE International Conference on Multimedia and Expo (ICME)*, pages 1452–1457. IEEE, 2017.

[50] Martin Ester, Hans-Peter Kriegel, Jörg Sander, Xiaowei Xu, et al. A density-based algorithm for discovering clusters in large spatial databases with noise. In *Proceedings of the Second International Conference on Knowledge Discovery and Data Mining*, pages 226–231, 1996.

[51] Pierre Thomas Faure-Giovagnoli. Deep-learning for automated diatom detection and identification for the ecological diagnosis of freshwater environments. Master's thesis, Georgia Institute of Technology, 2020.

[52] Stanislav Fort, Jie Ren, and Balaji Lakshminarayanan. Exploring the limits of out-of-distribution detection. *Advances in Neural Information Processing Systems*, 34:7068–7081, 2021.

[53] Andrea Frome, Greg S Corrado, Jon Shlens, Samy Bengio, Jeff Dean, Marc'Aurelio Ranzato, and Tomas Mikolov. Devise: A deep visual-semantic embedding model. In *Advances in neural information processing systems*, pages 2121–2129, 2013.

[54] Kunihiko Fukushima. Cognitron: A self-organizing multilayered neural network. *Biological cybernetics*, 20:121–136, 1975.

[55] Michael Gadermayr, Andreas Uhl, and Andreas Vécsei. Dealing with intra-class and intra-image variations in automatic celiac disease diagnosis. In *Bildverarbeitung für die Medizin 2015*, pages 461–466. Springer, 2015.

[56] Yarin Gal. *Uncertainty in deep learning*. PhD thesis, University of Cambridge, 2016.

[57] Yarin Gal and Zoubin Ghahramani. Dropout as a bayesian approximation: Representing model uncertainty in deep learning. In *international conference on machine learning*, pages 1050–1059. PMLR, 2016.

[58] Jakob Gawlikowski, Cedrique Rovile Njieutcheu Tassi, Mohsin Ali, Jongseok Lee, Matthias Humt, Jianxiang Feng, Anna Kruspe, Rudolph Triebel, Peter Jung, Ribana Roscher, et al. A survey of uncertainty in deep neural networks. *arXiv preprint arXiv:2107.03342*, 2021.

[59] Adas Gelzinis, Antanas Verikas, Evaldas Vaiciukynas, and Marija Bacauskiene. A novel technique to extract accurate cell contours applied for segmentation of phytoplankton images. *Machine Vision and Applications*, 26:305–315, 2015.

[60] Golnaz Ghiasi, Yin Cui, Aravind Srinivas, Rui Qian, Tsung-Yi Lin, Ekin D Cubuk, Quoc V Le, and Barret Zoph. Simple copy-paste is a strong data augmentation method for instance segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2918–2928, 2021.

[61] Ross Girshick. Fast R-CNN. In *Proceedings of the IEEE international conference on computer vision*, pages 1440–1448, 2015.

[62] Ross Girshick, Jeff Donahue, Trevor Darrell, and Jitendra Malik. Rich feature hierarchies for accurate object detection and semantic segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 580–587, 2014.

[63] Ethan Goan and Clinton Fookes. Bayesian neural networks: An introduction and survey. In *Case Studies in Applied Bayesian Data Science*, pages 45–87. Springer, 2020.

[64] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. *Advances in neural information processing systems*, 27:2672–2680, 2014.

[65] Brent Griffin. Mobile robot manipulation using pure object detection. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 561–571, 2023.

[66] Sorin Grigorescu, Bogdan Trasnea, Tiberiu Cocias, and Gigel Macesanu. A survey of deep learning techniques for autonomous driving. *Journal of Field Robotics*, 37:362–386, 2020.

[67] Chuan Guo, Geoff Pleiss, Yu Sun, and Kilian Q Weinberger. On calibration of modern neural networks. In *International conference on machine learning*, pages 1321–1330. PMLR, 2017.

[68] Peter Haase, John Murray-Bligh, Susanne Lohse, Steffen Pauls, Andrea Sundermann, Rick Gunn, and Ralph Clarke. Assessing the impact of errors in sorting and identifying macroinvertebrate samples. *The Ecological Status of European Rivers: Evaluation and Intercalibration of Assessment Methods*, pages 505–521, 2006.

[69] Trevor Hastie, Robert Tibshirani, Jerome H Friedman, and Jerome H Friedman. The elements of statistical learning: data mining, inference, and prediction. *Springer Series in Statistics*, 2:1–764, 2009.

[70] Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross Girshick. Mask r-cnn. In *Proceedings of the IEEE international conference on computer vision*, pages 2961–2969, 2017.

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

[72] Wenchong He and Zhe Jiang. A survey on uncertainty quantification methods for deep neural networks: An uncertainty source perspective. *arXiv preprint arXiv:2302.13425*, 2023.

[73] Dan Hendrycks and Kevin Gimpel. A baseline for detecting misclassified and out-of-distribution examples in neural networks. *arXiv preprint arXiv:1610.02136*, 2016.

[74] Andrew G Howard, Menglong Zhu, Bo Chen, Dmitry Kalenichenko, Weijun Wang, Tobias Weyand, Marco Andreetto, and Hartwig Adam. Mobilenets: Efficient convolutional neural networks for mobile vision applications. *arXiv preprint arXiv:1704.04861*, 2017.

[75] Ming-Kuei Hu. Visual pattern recognition by moment invariants. *IRE transactions on information theory*, 8:179–187, 1962.

[76] Qiao Hu and Cabell Davis. Automatic plankton image recognition with co-occurrence matrices and support vector machine. *Marine Ecology Progress Series*, 295:21–31, 2005.

[77] Gao Huang, Zhuang Liu, Laurens Van Der Maaten, and Kilian Q Weinberger. Densely connected convolutional networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4700–4708, 2017.

[78] Eyke Hüllermeier and Willem Waegeman. Aleatoric and epistemic uncertainty in machine learning: An introduction to concepts and methods. *Machine Learning*, 110:457–506, 2021.

[79] Francois Husson, Julie Josse, Sebastien Le, Jeremy Mazet, and Maintainer Francois Husson. Package 'factominer'. *An R package*, 96:698, 2016.

[80] Licheng Jiao, Fan Zhang, Fang Liu, Shuyuan Yang, Lingling Li, Zhixi Feng, and Rong Qu. A survey of deep learning-based object detection. *IEEE Access*, 7:128837–128868, 2019.

[81] Glenn Jocher, Alex Stoken, Jirka Borovec, NanoCode012, ChristopherSTAN, Liu Changyu, Laughing, tkianai, Adam Hogan, lorenzomammana, yxNONG, AlexWang1900, Laurentiu Diaconu, Marc, wanghaoyang0106, ml5ah, Doug, Francisco Ingham, Frederik, Guilhen, Hatovix, Jake Poznanski, Jiacong Fang, Lijun Yu , changyu98, Mingyu Wang, Naman Gupta, Osama Akhtar, PetrDvoracek, and Prashant Rai. ultralytics/yolov5: v3.1 - Bug Fixes and Performance Improvements, October 2020.

[82] Jeff Johnson, Matthijs Douze, and Hervé Jégou. Billion-scale similarity search with gpus. *IEEE Transactions on Big Data*, 7:535–547, 2019.

[83] Laurent Valentin Jospin, Hamid Laga, Farid Boussaid, Wray Buntine, and Mohammed Bennamoun. Hands-on bayesian neural networks—a tutorial for deep learning users. *IEEE Computational Intelligence Magazine*, 17:29–48, 2022.

[84] Ryo Kamoi and Kei Kobayashi. Why is the mahalanobis distance effective for anomaly detection? *arXiv preprint arXiv:2003.00402*, 2020.

[85] Michael Kass, Andrew Witkin, and Demetri Terzopoulos. Snakes: Active contour models. *International journal of computer vision*, 1:321–331, 1988.

[86] Alboukadel Kassambara and Fabian Mundt. Package 'factoextra'. *Extract and visualize the results of multivariate data analyses*, 76, 2017.

[87] MG Kelly and Brian ALAN Whitton. The trophic diatom index: a new index for monitoring eutrophication in rivers. *Journal of applied phycology*, 7:433–444, 1995.

[88] Daria Kern and Andre Mastmeyer. 3d bounding box detection in volumetric medical image data: A systematic literature review. In *2021 IEEE 8th International Conference on Industrial Engineering and Applications (ICIEA)*, pages 509–516. IEEE, 2021.

[89] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

[90] Michael Kloster, Andrea M Burfeid-Castellanos, Daniel Langenkämper, Tim W Nattkemper, and Bánk Beszteri. Improving deep learning-based segmentation of diatoms in gigapixel-sized virtual slides by object-based tile positioning and object integrity constraint. *Plos one*, 18:e0272103, 2023.

[91] Michael Kloster, Gerhard Kauer, and Bánk Beszteri. Sherpa: an image segmentation and outline feature extraction tool for diatoms and other objects. *BMC bioinformatics*, 15:1–17, 2014.

[92] Michael Kloster, Daniel Langenkämper, Martin Zurowietz, Bánk Beszteri, and Tim W Nattkemper. Deep learning-based diatom taxonomy on virtual slides. *Scientific reports*, 10:14416, 2020.

[93] Alex Krizhevsky and Geoffrey Hinton. Learning multiple layers of features from tiny images. Technical report, University of Toronto, Toronto, Ontario, 2009.

[94] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. *Communications of the ACM*, 60:84–90, 2017.

[95] Balaji Lakshminarayanan, Alexander Pritzel, and Charles Blundell. Simple and scalable predictive uncertainty estimation using deep ensembles. *Advances in neural information processing systems*, 30:6405–6416, 2017.

[96] Christian Lalanne-Cassou and Jean-François Voisin. Atlas des diatomées d'Île de France. pages 1–734. Direction Régionale et Interdépartmentale de l'Environnement et de l'Energie d'Île-de-France, 2013.

[97] Floriane Larras, Romain Coulaud, Edwige Gautreau, Elise Billoir, Juliette Rosebery, and Philippe Usseglio-Polatera. Assessing anthropogenic pressures on streams: A random forest approach based on benthic diatom communities. *Science of the Total Environment*, 586:1101–1112, 2017.

[98] Floriane Larras and Philippe Usseglio-Polatera. Heterogeneity in macroinvertebrate sampling strategy introduces variability in community characterization and stream trait-based biomonitoring: Influence of sampling effort and habitat selection criteria. *Ecological Indicators*, 119:106758, 2020.

[99] Mans Larsson, Erik Stenborg, Carl Toft, Lars Hammarstrand, Torsten Sattler, and Fredrik Kahl. Fine-grained segmentation networks: Self-supervised segmentation for improved long-term visual localization. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 31–41, 2019.

[100] Isabelle Lavoie, Paul B Hamilton, Soizic Morin, Sandra Kim Tiam, Maria Kahlert, Sara Gonçalves, Elisa Falasco, Claude Fortin, Brigitte Gontero, David Heudre, Mila Kojadinovic-Sirinelli, Kalina Manoylov, Lalit K Pandey, and Jonathan C Taylor. Diatom teratologies as biomarkers of contamination: are all deformities ecologically meaningful? *Ecological Indicators*, 82:539–550, 2017.

[101] Yann LeCun, Léon Bottou, Yoshua Bengio, and Patrick Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86:2278–2324, 1998.

[102] Yann LeCun, Corinna Cortes, and CJ Burges. Mnist handwritten digit database. *ATT Labs [Online]. Available: http://yann.lecun.com/exdb/mnist*, 2, 2010.

[103] Hansang Lee, Minseok Park, and Junmo Kim. Plankton classification on imbalanced large scale database via convolutional neural networks with transfer learning. In *2016 IEEE international conference on image processing*, pages 3713–3717. IEEE, 2016.

[104] Kimin Lee, Kibok Lee, Honglak Lee, and Jinwoo Shin. A simple unified framework for detecting out-of-distribution samples and adversarial attacks. *Advances in neural information processing systems*, 31:7167–7177, 2018.

[105] Sylvia S Lee, Ian W Bishop, Sarah A Spaulding, Richard M Mitchell, and Lester L Yuan. Taxonomic harmonization may reveal a stronger association between diatom assemblages and total phosphorus in large datasets. *Ecological indicators*, 102:166–174, 2019.

[106] Fabien Leprieur, Olivier Beauchard, Simon Blanchet, Thierry Oberdorff, and Sébastien Brosse. Fish invasions in the world's river systems: when natural processes are blurred by human activities. *PLoS biology*, 6:e28, 2008.

[107] Linhao Li, Zhiqiang Zhou, Bo Wang, Lingjuan Miao, and Hua Zong. A novel CNN-based method for accurate ship detection in HR optical remote sensing images via rotated bounding box. *IEEE Transactions on Geoscience and Remote Sensing*, 59:686–699, 2020.

[108] Shuxin Li, Zhilong Zhang, Biao Li, and Chuwei Li. Multiscale rotated bounding box-based deep learning method for detecting ship targets in remote sensing images. *Sensors*, 18:2702, 2018.

[109] Shiyu Liang, Yixuan Li, and Rayadurgam Srikant. Enhancing the reliability of out-of-distribution image detection in neural networks. *arXiv preprint arXiv:1706.02690*, 2017.

[110] Tsung-Yi Lin, Priya Goyal, Ross Girshick, Kaiming He, and Piotr Dollár. Focal loss for dense object detection. In *Proceedings of the IEEE international conference on computer vision*, pages 2980–2988, 2017.

[111] Geert Litjens, Thijs Kooi, Babak Ehteshami Bejnordi, Arnaud Arindra Adiyoso Setio, Francesco Ciompi, Mohsen Ghafoorian, Jeroen Awm Van Der Laak, Bram Van Ginneken, and Clara I Sánchez. A survey on deep learning in medical image analysis. *Medical image analysis*, 42:60–88, 2017.

[112] Jeremiah Liu, Zi Lin, Shreyas Padhy, Dustin Tran, Tania Bedrax Weiss, and Balaji Lakshminarayanan. Simple and principled uncertainty estimation with deterministic deep learning via distance awareness. *Advances in Neural Information Processing Systems*, 33:7498–7512, 2020.

[113] Wei Liu, Dragomir Anguelov, Dumitru Erhan, Christian Szegedy, Scott Reed, Cheng-Yang Fu, and Alexander C Berg. SSD: Single shot multibox detector. In *14th European Conference on Computer Vision*, pages 21–37. Springer, 2016.

[114] Weitang Liu, Xiaoyun Wang, John Owens, and Yixuan Li. Energy-based out-of-distribution detection. *Advances in Neural Information Processing Systems*, 33:21464–21475, 2020.

[115] Stuart Lloyd. Least squares quantization in PCM. *IEEE transactions on information theory*, 28:129–137, 1982.

[116] Jonathan Long, Evan Shelhamer, and Trevor Darrell. Fully convolutional networks for semantic segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3431–3440, 2015.

[117] David G Lowe. Object recognition from local scale-invariant features. In *Proceedings of the seventh IEEE international conference on computer vision*, pages 1150–1157. IEEE, 1999.

[118] Alessandra Lumini and Loris Nanni. Deep learning and transfer learning features for plankton classification. *Ecological informatics*, 51:33–43, 2019.

[119] Alessandra Lumini, Loris Nanni, and Gianluca Maguolo. Deep learning for plankton and coral classification. *Applied Computing and Informatics*, 19:265–283, 2020.

[120] Prasanta Chandra Mahalanobis. On test and measures of group divergence. *Journal of Asiatic Society of Bengal*, 26:541–588, 1930.

[121] Andrey Malinin and Mark Gales. Predictive uncertainty estimation via prior networks. *Advances in neural information processing systems*, 31:7047–7058, 2018.

[122] David G Mann. The species concept in diatoms. *Phycologia*, 38:437–495, 1999.

[123] Séverine Martini, Floriane Larras, Aurélien Boyé, Emile Faure, Nicole Aberle, Philippe Archambault, Lise Bacouillard, Beatrix E Beisner, Lucie Bittner, Emmanuel Castella, Michael Danger, Olivier Gauthier, Lee Karp-Boss, Fabien Lombard, Frédéric Maps, Lars Stemmann, Eric Thiébaut, Philippe Usseglio-Polatera, Meike Vogt, Martin Laviale, and Sakina-Dorothée Ayata. Functional trait-based approaches as a common framework for aquatic ecologists. *Limnology and Oceanography*, 66:965–994, 2021.

[124] Anahita Marzin, Olivier Delaigue, Maxime Logez, Jérôme Belliard, and Didier Pont. Uncertainty associated with river health assessment in a varying environment: The case of a predictive fish-based index in france. *Ecological Indicators*, 43:195–204, 2014.

[125] George A Miller. Wordnet: a lexical database for english. *Communications of the ACM*, 38:39–41, 1995.

[126] George A Miller. *WordNet: An electronic lexical database*. MIT press, 1998.

[127] Shervin Minaee, Yuri Boykov, Fatih Porikli, Antonio Plaza, Nasser Kehtarnavaz, and Demetri Terzopoulos. Image segmentation using deep learning: A survey. *IEEE transactions on pattern analysis and machine intelligence*, 44:3523–3542, 2021.

[128] Soizic Morin. *Bioindication des effets des pollutions métalliques sur les communautés de diatomées benthiques. Approches in situ et expérimentales*. PhD thesis, Université Sciences et Technologies-Bordeaux I, 2006.

[129] Laurent Najman and Michel Schmitt. Watershed of a continuous function. *Signal Processing*, 38:99–112, 1994.

[130] Yuval Netzer, Tao Wang, Adam Coates, Alessandro Bissacco, Bo Wu, and Andrew Y. Ng. Reading digits in natural images with unsupervised feature learning. In *NIPS Workshop on Deep Learning and Unsupervised Feature Learning 2011*, 2011.

[131] Richard Nock and Frank Nielsen. Statistical region merging. *IEEE transactions on pattern analysis and machine intelligence*, 26:1452–1458, 2004.

[132] Eric C Orenstein, Sakina-Dorothée Ayata, Frédéric Maps, Érica C Becker, Fabio Benedetti, Tristan Biard, Thibault de Garidel-Thoron, Jeffrey S Ellen, Filippo Ferrario, Sarah LC Giering, et al. Machine learning techniques to characterize functional traits of plankton from image data. *Limnology and oceanography*, 67:1647–1669, 2022.

[133] Eric C Orenstein, Oscar Beijbom, Emily E Peacock, and Heidi M Sosik. Whoi-plankton-a large scale fine grained visual recognition benchmark dataset for plankton classification. *arXiv preprint arXiv:1510.00745*, 2015.

[134] Nobuyuki Otsu. A threshold selection method from gray-level histograms. *IEEE transactions on systems, man, and cybernetics*, 9:62–66, 1979.

[135] Ingrid D Pardo, Robert H Garman, Klaus Weber, Walter F Bobrowski, Jerry F Hardisty, and Daniel Morton. Technical guide for nervous system sampling of the cynomolgus monkey for general toxicity studies. *Toxicologic pathology*, 40:624–636, 2012.

[136] Gunhan Park, Yunju Baek, and Heung-Kyu Lee. A ranking algorithm using dynamic clustering for content-based image retrieval. In *Proceedings of the International Conferece on Image and Video Retrieval*, pages 328–337. Springer, 2002.

[137] Vito P Pastore, Thomas G Zimmerman, Sujoy K Biswas, and Simone Bianco. Annotation-free learning of plankton for classification and anomaly detection. *Scientific reports*, 10:1–15, 2020.

[138] Anibal Pedraza, Gloria Bueno, Oscar Deniz, Gabriel Cristóbal, Saúl Blanco, and María Borrego-Ramos. Automated diatom classification (Part B): a deep learning approach. *Applied Sciences*, 7:460, 2017.

[139] Valérie Peeters and Luc Ector. Atlas des diatomées des cours d'eau du territoire bourguignon. volume 2: Monoraphidées, brachyraphidées. pages 1–271. Direction Régionale de l'Environnement, de l'Aménagement et du Logement, Bourgogne-Franche-Comté. Dijon, 2018.

[140] Dan Pelleg, Andrew W Moore, et al. X-means: Extending k-means with efficient estimation of the number of clusters. In *International Conference on Machine Learning*, volume 1, pages 727–734, 2000.

[141] Patrick Pérez, Michel Gangnet, and Andrew Blake. Poisson image editing. In *ACM SIGGRAPH 2003 Papers*, pages 313–318. 2003.

[142] Rafał Pilarczyk and Władysław Skarbek. On intra-class variance for deep learning of classifiers. *Foundations of Computing and Decision Sciences*, 44:285–301, 2019.

[143] J Piper and T Piper. Microscopic modalities and illumination techniques. In *Modern Trends in Diatom Identification: Fundamentals and Applications*, pages 53–93. Springer, 2020.

[144] Janis Postels, Hermann Blum, Cesar Cadena, Roland Siegwart, Luc Van Gool, and Federico Tombari. Quantifying aleatoric and epistemic uncertainty using density estimation in latent space. *arXiv preprint arXiv:2012.03082*, 2020.

[145] Ouyang Py, Hu Hong, and Shi Zhongzhi. Plankton classification with deep convolutional neural networks. In *2016 IEEE Information Technology, Networking, Electronic and Automation Control Conference*, pages 132–136. IEEE, 2016.

[146] Yuan Qi, Xihong Lian, Hongwei Wang, Jinlong Zhang, and Rui Yang. Dynamic mechanism between human activities and ecosystem services: A case study of qinghai lake watershed, china. *Ecological Indicators*, 117:106528, 2020.

[147] Wen Qian, Xue Yang, Silong Peng, Junchi Yan, and Yue Guo. Learning modulated loss for rotated object detection. In *Proceedings of the AAAI conference on artificial intelligence*, volume 35, pages 2458–2466, 2021.

[148] Joseph Redmon, Santosh Divvala, Ross Girshick, and Ali Farhadi. You only look once: Unified, real-time object detection. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 779–788, 2016.

[149] Jie Ren, Stanislav Fort, Jeremiah Liu, Abhijit Guha Roy, Shreyas Padhy, and Balaji Lakshminarayanan. A simple fix to mahalanobis distance for improving near-ood detection. *arXiv preprint arXiv:2106.09022*, 2021.

[150] Pengzhen Ren, Yun Xiao, Xiaojun Chang, Po-Yao Huang, Zhihui Li, Brij B Gupta, Xiaojiang Chen, and Xin Wang. A survey of deep active learning. *ACM computing surveys*, 54:1–40, 2021.

[151] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster R-CNN: Towards real-time object detection with region proposal networks. *Advances in neural information processing systems*, 28:91–99, 2015.

[152] Yorick Reyjol, Christine Argillier, Wendy Bonne, Angel Borja, Anthonie D Buijse, Ana Cristina Cardoso, Martin Daufresne, Martin Kernan, Maria Teresa Ferreira, Sandra Poikane, et al. Assessing the ecological status in the context of the european water framework directive: where do we go now? *Science of the Total Environment*, 497:332–344, 2014.

[153] Francisco Caio Maia Rodrigues, Nina ST Hirata, Antonio A Abello, T Leandro, De La Cruz, Rubens M Lopes, and Roberto Hirata Jr. Evaluation of transfer learning scenarios in plankton image classification. In *Proceedings of International Joint Conference on Computer Vision, Imaging and Computer Graphics*, pages 359–366, 2018.

[154] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. In *Proceedings of 18th International Conference on Medical Image Computing and Computer-Assisted Intervention*, pages 234–241. Springer, 2015.

[155] Jesus Ruiz-Santaquiteria, Gloria Bueno, Oscar Deniz, Noelia Vallez, and Gabriel Cristobal. Semantic versus instance segmentation in microscopic algae detection. *Engineering Applications of Artificial Intelligence*, 87:103271, 2020.

[156] Christoph Sager, Christian Janiesch, and Patrick Zschech. A survey of image labelling for computer vision applications. *Journal of Business Analytics*, 4:91–110, 2021.

[157] Jesús Salido, Carlos Sánchez, Jesús Ruiz-Santaquiteria, Gabriel Cristóbal, Saul Blanco, and Gloria Bueno. A low-cost automated digital microscopy platform for automatic identification of diatoms. *Applied Sciences*, 10:6033, 2020.

[158] Caroline A Schneider, Wayne S Rasband, and Kevin W Eliceiri. NIH image to ImageJ: 25 years of image analysis. *Nature methods*, 9:671–675, 2012.

[159] Florian Schroff, Dmitry Kalenichenko, and James Philbin. FaceNet: A unified embedding for face recognition and clustering. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 815–823, 2015.

[160] Murat Sensoy, Lance Kaplan, and Melih Kandemir. Evidential deep learning to quantify classification uncertainty. *Advances in neural information processing systems*, 31:3183–3193, 2018.

[161] João Serôdio and Johann Lavaud. Diatoms and their ecological importance. In *Life Below Water*, pages 304–312. Springer, 2022.

[162] Burr Settles. From theories to queries: Active learning in practice. In *Active learning and experimental design workshop in conjunction with AISTATS*, pages 1–18. JMLR Workshop and Conference Proceedings, 2011.

[163] Carlos N Silla and Alex A Freitas. A survey of hierarchical classification across different application domains. *Data Mining and Knowledge Discovery*, 22:31–72, 2011.

[164] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014.

[165] Lewis Smith and Yarin Gal. Understanding measures of uncertainty for adversarial example detection. *arXiv preprint arXiv:1803.08533*, 2018.

[166] Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. Dropout: a simple way to prevent neural networks from overfitting. *The journal of machine learning research*, 15:1929–1958, 2014.

[167] Christian Szegedy, Wei Liu, Yangqing Jia, Pierre Sermanet, Scott Reed, Dragomir Anguelov, Dumitru Erhan, Vincent Vanhoucke, and Andrew Rabinovich. Going deeper with convolutions. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1–9, 2015.

[168] Mengyu Tan, Wentao Chao, Jo-Ku Cheng, Mo Zhou, Yiwen Ma, Xinyi Jiang, Jianping Ge, Lian Yu, and Limin Feng. Animal detection and classification from camera trap images using different mainstream object detection architectures. *Animals*, 12:1976, 2022.

[169] Mingxing Tan and Quoc Le. Efficientnet: Rethinking model scaling for convolutional neural networks. In *International conference on machine learning*, pages 6105–6114. PMLR, 2019.

[170] Ning Tang, Fei Zhou, Zhaorui Gu, Haiyong Zheng, Zhibin Yu, and Bing Zheng. Unsupervised pixel-wise classification for chaetoceros image segmentation. *Neurocomputing*, 318:261–270, 2018.

[171] Xiaoou Tang, W Kenneth Stewart, He Huang, Scott M Gallager, Cabell S Davis, Luc Vincent, and Marty Marra. Automatic plankton image recognition. *Artificial intelligence review*, 12:177–199, 1998.

[172] Hugo Touvron, Matthieu Cord, Matthijs Douze, Francisco Massa, Alexandre Sablayrolles, and Hervé Jégou. Training data-efficient image transformers & distillation

through attention. In *International conference on machine learning*, pages 10347–10357. PMLR, 2021.

[173] Robert L Vadas Jr, Robert M Hughes, Yeon Jae Bae, Min Jeong Baek, Orestes Carlos Bello Gonzáles, Marcos Callisto, Débora Reis de Carvalho, Kai Chen, Maria T Ferreira, Pablo Fierro, Jon S. Harding, Dana M Infante, C.J Kleynhans, Diego R Macedo, Isabela Martins, Norman Mercado Silva, Nabor Moya, Susan J. Nichols, Paulo S. Pompeu, Renata Ruaro, Deborah R.O. Silva, R. Jan Stevenson, Bianca de Freitas Terra, Christa Thirion, Douglas Ticiani, Lizhu Wang, and Chris O. Yoder. Assemblage-based biomonitoring of freshwater ecosystem health via multimetric indices: A critical review and suggestions for improving their applicability. *Water Biology and Security*, 1:100054, 2022.

[174] Noelia Vallez, Gloria Bueno, Oscar Deniz, and Saul Blanco. Diffeomorphic transforms for data augmentation of highly variable shape and texture objects. *Computer Methods and Programs in Biomedicine*, 219:106775, 2022.

[175] Joost van Amersfoort, Lewis Smith, Andrew Jesson, Oscar Key, and Yarin Gal. On feature collapse and deep kernel learning for single forward pass uncertainty. *arXiv preprint arXiv:2102.11409*, 2021.

[176] Joost Van Amersfoort, Lewis Smith, Yee Whye Teh, and Yarin Gal. Uncertainty estimation using a single deep deterministic neural network. In *International conference on machine learning*, pages 9690–9700. PMLR, 2020.

[177] Laurens Van der Maaten and Geoffrey Hinton. Visualizing data using t-SNE. *Journal of machine learning research*, 9:2579–2605, 2008.

[178] Garnik Vareldzhan, Kirill Yurkov, and Konstantin Ushenin. Anomaly detection in image datasets using convolutional neural networks, center loss, and mahalanobis distance. In *2021 Ural Symposium on Biomedical Engineering, Radioelectronics and Information Technology*, pages 387–390. IEEE, 2021.

[179] Aishwarya Venkataramanan, Assia Benbihi, Martin Laviale, and Cédric Pradalier. Gaussian latent representations for uncertainty estimation using mahalanobis distance in deep classifiers. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 4488–4497, 2023.

[180] Aishwarya Venkataramanan, Pierre Faure-Giovagnoli, Cyril Regan, David Heudre, Cécile Figus, Philippe Usseglio-Polatera, Cedric Pradalier, and Martin Laviale. Use-

fulness of synthetic datasets for diatom automatic detection using a deep-learning approach. *Engineering Applications of Artificial Intelligence*, 117:105594, 2023.

[181] Aishwarya Venkataramanan, Martin Laviale, Cécile Figus, Philippe Usseglio-Polatera, and Cédric Pradalier. Tackling inter-class similarity and intra-class variance for microscopic image-based classification. In *International conference on computer vision systems*, pages 93–103. Springer, 2021.

[182] Aishwarya Venkataramanan, Martin Laviale, and Cédric Pradalier. Integrating visual and semantic similarity using hierarchies for image retrieval. In *International Conference on Computer Vision Systems*, pages 422–431. Springer, 2023.

[183] Antanas Verikas, Adas Gelzinis, Marija Bacauskiene, Irina Olenina, Sergej Olenin, and Evaldas Vaiciukynas. Phase congruency-based detection of circular objects applied to analysis of phytoplankton images. *Pattern Recognition*, 45:1659–1670, 2012.

[184] Paul Viola and Michael Jones. Rapid object detection using a boosted cascade of simple features. In *Proceedings of the 2001 IEEE computer society conference on computer vision and pattern recognition*, volume 1. IEEE, 2001.

[185] Marie Wach, Julie Guéguen, Christian Chauvin, François Delmas, Nina Dagens, Thibaut Feret, Sandrine Loriot, and Juliette Tison-Rosebery. Probability of misclassifying river ecological status: A large-scale approach to assign uncertainty in macrophyte and diatom-based biomonitoring. *Ecological Indicators*, 101:285–295, 2019.

[186] Catherine Wah, Steve Branson, Peter Welinder, Pietro Perona, and Serge Belongie. The caltech-ucsd birds-200-2011 dataset. Technical Report CNS-TR-2011-001, California Institute of Technology, 2011.

[187] Ji Wan, Dayong Wang, Steven Chu Hong Hoi, Pengcheng Wu, Jianke Zhu, Yongdong Zhang, and Jintao Li. Deep learning for content-based image retrieval: A comprehensive study. In *Proceedings of the 22nd ACM international conference on Multimedia*, pages 157–166, 2014.

[188] Chao Wang, Zhibin Yu, Haiyong Zheng, Nan Wang, and Bing Zheng. Cgan-plankton: Towards large-scale imbalanced class generation and fine-grained classification. In *2017 IEEE International Conference on Image Processing (ICIP)*, pages 855–859. IEEE, 2017.

[189] JG Wasson, A Chandesris, H Pella, and L Blanc. *Définition des hydro-écorégions françaises métropolitaines. Approche régionale de la typologie des eaux courantes et*

éléments pour la définition des peuplements de référence d'invertébrés. PhD thesis, irstea, 2002.

[190] Juliane Wiederkehr, Corinne Grac, Mickaël Fabrègue, Bruno Fontan, Frédéric Labat, Florence Le Ber, and Michèle Trémolières. Experimental study of uncertainties on the macrophyte index (IBMR) based on species identification and cover. *Ecological Indicators*, 50:242–250, 2015.

[191] Juliane Wiederkehr, Corinne Grac, Bruno Fontan, Frédéric Labat, Florence Le Ber, and Michèle Trémolières. Experimental study of the uncertainty of the intrasubstrate variability on two french index metrics based on macroinvertebrates. *Hydrobiologia*, 779:59–73, 2016.

[192] Jim Winkens, Rudy Bunel, Abhijit Guha Roy, Robert Stanforth, Vivek Natarajan, Joseph R Ledsam, Patricia MacWilliams, Pushmeet Kohli, Alan Karthikesalingam, Simon Kohl, Taylan Cemgil, S. M. Ali Eslami, and Olaf Ronneberger. Contrastive training for improved out-of-distribution detection. *arXiv preprint arXiv:2007.05566*, 2020.

[193] Han Xiao, Kashif Rasul, and Roland Vollgraf. Fashion-MNIST: a novel image dataset for benchmarking machine learning algorithms. *arXiv preprint arXiv:1708.07747*, 2017.

[194] Zhicheng Yan, Hao Zhang, Robinson Piramuthu, Vignesh Jagadeesh, Dennis DeCoste, Wei Di, and Yizhou Yu. HD-CNN: hierarchical deep convolutional neural networks for large scale visual recognition. In *Proceedings of the IEEE international conference on computer vision*, pages 2740–2748, 2015.

[195] Jingkang Yang, Kaiyang Zhou, Yixuan Li, and Ziwei Liu. Generalized out-of-distribution detection: A survey. *arXiv preprint arXiv:2110.11334*, 2021.

[196] Jufeng Yang, Dongyu She, Yu-Kun Lai, and Ming-Hsuan Yang. Retrieving and classifying affective images via deep metric learning. In *Proceedings of the AAAI Conference*, volume 32, 2018.

[197] Lu Yang, He Jiang, Qing Song, and Jun Guo. A survey on long-tailed visual recognition. *International Journal of Computer Vision*, 130:1837–1872, 2022.

[198] Xue Yang, Junchi Yan, Qi Ming, Wentao Wang, Xiaopeng Zhang, and Qi Tian. Rethinking rotated object detection with gaussian wasserstein distance loss. In *International Conference on Machine Learning*, pages 11830–11841. PMLR, 2021.

[199] Xue Yang, Xiaojiang Yang, Jirui Yang, Qi Ming, Wentao Wang, Qi Tian, and Junchi Yan. Learning high-precision bounding box for rotated object detection via kullback-leibler divergence. *Advances in Neural Information Processing Systems*, 34:18381–18394, 2021.

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

[201] Tianlin Zhai, Jing Wang, Ying Fang, Yun Qin, Longyang Huang, and Ye Chen. Assessing ecological risks caused by human activities in rapid urbanization coastal areas: Towards an integrated approach to determining key areas of terrestrial-oceanic ecosystems preservation and restoration. *Science of the Total Environment*, 708:135153, 2020.

[202] Chunjie Zhang, Jian Cheng, and Qi Tian. Image-level classification by hierarchical structure learning with visual and semantic similarities. *Information Sciences*, 422:271–281, 2018.

[203] Yifan Zhang, Bingyi Kang, Bryan Hooi, Shuicheng Yan, and Jiashi Feng. Deep long-tailed learning: A survey. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2023.

[204] Feng Zhao, Feng Lin, and Hock Soon Seah. Binary sipper plankton image classification using random subspace. *Neurocomputing*, 73:1853–1860, 2010.

[205] Hengshuang Zhao, Jianping Shi, Xiaojuan Qi, Xiaogang Wang, and Jiaya Jia. Pyramid scene parsing network. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2881–2890, 2017.

[206] Zhonghua Zhao, Shanqing Guo, Qiuliang Xu, and Tao Ban. G-means: a clustering algorithm for intrusion detection. In *15th International Conference on Advances in Neuro-Information Processing, Revised Selected Papers, Part I*, pages 563–570. Springer, 2009.

[207] Haiyong Zheng, Ruchen Wang, Zhibin Yu, Nan Wang, Zhaorui Gu, and Bing Zheng. Automatic plankton image classification combining multiple view features via multiple kernel learning. *BMC bioinformatics*, 18:1–18, 2017.

[208] Bo Zhong and Kai Ao. Single-stage rotation-decoupled detector for oriented object. *Remote Sensing*, 12:3262, 2020.

[209] Matteo Zucchetta, Luca Scapin, Anita Franco, and Piero Franzoi. Uncertainty in developing fish based multi-metric indices. *Ecological Indicators*, 108:105768, 2020.

Appendix A

Atlas Dataset

Table A.1: List of taxa used for training the classifier

Code	Taxa Name	Number of Images
AAMB	Aulacoseira ambigua (Grunow) Simonsen	142
ACAF	Achnanthidium affine (Grun) Czarnecki	70
ACLI	Achnanthidium lineare W, Smith	185
ACOF	Amphora coffeaeformis (Agardh) Kützing var, coffeaeformis	30
ACOP	Amphora copulata (Kützing) Schoeman et Archibald var, copulata	32
ADCT	Achnanthidium catenatum (Bily et Marvan) Lange-Bertalot	209
ADEG	Achnanthidium exiguum (Grunow) Czarnecki var, exiguum	87
ADEU	Achnanthidium eutrophilum (Lange-Bertalot)Lange-Bertalot	235
ADGL	Achnanthidium gracillimum (Meister)Lange-Bertalot	44
ADKR	Achnanthidium kranzii (Lange-Bertalot) Round et Bukhtiyarova	58
ADLA	Achnanthidium latecephalum Kobayasi	61

Continued on next page

Table A.1 – *Continued*

Code	Taxa Name	Number of Images
ADMI	Achnanthidium minutissimum (Kützing) Czarnecki var, minutissimum	398
ADMS	Adlafia minuscula (Grunow) Lange-Bertalot var, minuscula	39
ADPY	Achnanthidium pyrenaicum (Hustedt) Kobayasi	179
ADRI	Achnanthidium rivulare Potapova et Ponader	63
ADSA	Achnanthidium saprophilum (Kobayasi et Mayama) Round et Bukhtiyarova	72
ADSB	Achnanthidium straubianum (Lange-Bertalot)Lange-Bertalot	504
ADSH	Achnanthidium subhudsonis (Hustedt) H, Kobayasi	75
ADSU	Achnanthidium subatomus (Hustedt) Lange-Bertalot var, subatomus	81
AFOR	Asterionella formosa Hassall var, formosa	94
AFUG	Achnanthes fugei Carter	39
AOVA	Amphora ovalis Lange-Bertalot var, ovalis	59
APED	Amphora pediculus (Kützing) Grunow var, pediculus	162
AUDI	Aulacoseira distans (Ehr,) Simonsen var, distans	42
AUGA	Aulacoseira granulata var, angustissima (O, Müller) Simonsen	75
AUGR	Aulacoseira granulata (Ehrenberg) Simonsen var, granulata	91
AUPU	Aulacoseira pusilla (Meister) Tuji et Houki	166
AUSU	Aulacoseira subarctica (O, Müller) Haworth var, subarctica	62
BNEO	Brachysira neoexilis Lange-Bertalot	37
CAEX	Cymbella excisa Kützing var, excisa	120
CAGR	Cyclotella atomus var, gracilis Genkal et Kiss	156
CATO	Cyclotella atomus Hustedt var, atomus	91
CBKU	Cymbopleura kuelbsii Krammer var, kuelbsii	33
CDTG	Cyclotella distinguenda Hustedt var, distinguenda	51

Continued on next page

Table A.1 – *Continued*

Code	Taxa Name	Number of Images
CDUB	Cyclostephanos dubius (Fricke) Round	147
CEUG	Cocconeis euglypta Ehrenberg	128
CINV	Cyclostephanos invisitatus Hohn et Hellerman)Theriot Stoermer et Håkansson	84
CLNT	Cocconeis lineata Ehrenberg	62
CMED	Cyclotella meduanae Germain	127
CMEN	Cyclotella meneghiniana Kützing var, meneghiniana	102
CMLF	Craticula molestiformis (Hustedt) Lange-Bertalot	41
CNDI	Cocconeis neodiminuta Krammer	43
CNLP	Cymbella neoleptoceros Krammer var, neoleptoceros	40
CNTH	Cocconeis neothumensis Krammer var, neothumensis	56
COCE	Cyclotella ocellata Pantocsek	96
COPL	Cocconeis pseudolineata (Geitler) Lange-Bertalot	123
CPED	Cocconeis pediculus Ehrenberg	65
CPLA	Cocconeis placentula Ehrenberg var, placentula	67
CPOL	Cyclotella polymorpha Meyer et Håkansson	33
CRAC	Craticula accomoda (Hustedt) D,G, Mann in Round et al,	81
CTUM	Cymbella tumida (Brébisson)Van Heurck var, tumida	33
DCOF	Diadesmis confervacea Kützing var, confervacea	79
DDEL	Delicata delicatula (Kützing) Krammer var, delicatula	31
DEHR	Diatoma ehrenbergii Kützing	40
DITE	Diatoma tenue Agardh var, tenue	66
DMES	Diatoma mesodon (Ehrenberg) Kützing	56
DMON	Diatoma moniliformis Kützing	66

Continued on next page

Table A.1 – *Continued*

Code	Taxa Name	Number of Images
DOBL	Diploneis oblongella (Naegeli) Cleve-Euler var, oblongella	31
DOCU	Diploneis oculata (Brébisson in Desmazières) Cleve	162
DPAR	Diploneis parma Cleve	72
DPST	Discostella pseudostelligera (Hustedt) Houk et Klee	71
DSTE	Discostella stelligera (Cleve et Grun,) Houk et Klee var, stelligera	70
DTEN	Denticula tenuis Kützing var, tenuis	133
DVUL	Diatoma vulgaris Bory var, vulgaris	82
ECAE	Encyonema caespitosum Kützing var, caespitosum	77
ECPM	Encyonopsis minuta Krammer et Reichardt	71
EEXI	Eunotia exigua (Brébisson ex Kützing) Rabenhorst var, exigua	69
EMIN	Eunotia minor (Kützing) Grunow in Van Heurck	44
ENCM	Encyonopsis microcephala (Grunow) Krammer var, microcephala	84
ENMI	Encyonema minutum (Hilse in Rabh,) D,G, Mann in Round Crawford et Mann var, minutum	85
ENNG	Encyonema neogracile Krammer var, neogracile	30
ENVE	Encyonema ventricosum (Kützing) Grunow in Schmidt et al, var, ventricosum	150
EOCO	Eolimna comperei Ector Coste et Iserentant in Coste et Ector	133
EOMI	Eolimna minima (Grunow) Lange-Bertalot	92
EORH	Eolimna rhombelliptica Moser Lange-Bertalot et Metzeltin	56
ESBM	Eolimna subminuscula (Manguin) Moser Lange-Bertalot et Metzeltin	109
ESLE	Encyonema silesiacum (Bleisch in Rabh,) D,G, Mann var, silesiacum	62
ESUM	Encyonopsis subminuta Krammer et Reichardt	53
ETEN	Eunotia tenella (Grunow in Van Heurck) Hustedt in Schmidt et al var, tenella	57
FAUT	Fragilaria austriaca (Grunow) Lange-Bertalot	47

Continued on next page

Table A.1 – *Continued*

Code	Taxa Name	Number of Images
FCRO	Fragilaria crotonensis Kitton var, crotonensis	50
FFVI	Fragilariforma virescens (Ralfs) Williams et Round var, virescens	34
FGRA	Fragilaria gracilis ostrup	112
FLEN	Fallacia lenzii (Hustedt) Lange-Bertalot in Werum et lange-Bertalot	91
FMES	Fragilaria mesolepta Rabenhorst	45
FMOC	Fallacia monoculata (Hustedt) D,G, Mann	47
FSAP	Fistulifera saprophila (Lange-Bertalot et Bonik) Lange-Bertalot	91
FSBH	Fallacia subhamulata (Grunow in V, Heurck) D,G, Mann	82
FSLU	Fallacia sublucidula (Hustedt) D,G, Mann	90
FVAU	Fragilaria vaucheriae (Kützing) Petersen var, vaucheriae	66
GACC	Geissleria acceptata (Hust,) Lange-Bertalot et Metzeltin	92
GBOB	Gomphonema bourbonense E, Reichardt et Lange-Bertalot	49
GCBC	Gomphonema cymbelliclinum Reichardt et Lange-Bertalot	38
GCLF	Gomphonema calcifugum Lange-Bertalot et Reichardt	50
GELG	Gomphonema elegantissimum Reichardt et Lange-Bertalot in Hofmann et al,	52
GLAT	Gomphonema lateripunctatum Reichardt et Lange-Bertalot	50
GLGN	Gomphonema lagenula Kützing	31
GMIC	Gomphonema micropus Kützing var, micropus	65
GMIN	Gomphonema minutum (Agardh) Agardh f, minutum	119
GMPU	Gomphonema micropumilum Reichardt	33
GOLD	Gomphonema olivaceoides Hustedt var, olivaceoides	43
GOLI	Gomphonema olivaceum (Hornemann) Brébisson var, olivaceum	83
GPAR	Gomphonema parvulum var, parvulum f, parvulum (Kützing) Kützing	91

Continued on next page

Table A.1 – *Continued*

Code	Taxa Name	Number of Images
GPEL	Gomphonema pumilum var, elegans Reichardt et Lange-Bertalot	46
GPLI	Gomphosphenia lingulatiformis (Lange-Bertalot et Reichardt) Lange-Bertalot	59
GPRI	Gomphonema pumilum var, rigidum Reichardt et Lange-Bertalot	120
GPRO	Gomphonema productum (Grunow) Lange-Bertalot et Reichardt	38
GPSA	Gomphonema pseudoaugur Lange-Bertalot	35
GTER	Gomphonema tergestinum (Grunow in Van Heurck) Schmidt in Schmidt et al, var, tergestinum	107
HACO	Halamphora coffeaeformis (Agardh) Levkov	30
HARC	Hannaea arcus (Ehr,) Patrick var, arcus	39
HCAP	Hippodonta capitata (Ehr,)Lange-Bertalot, Metzeltin et Witkowski	40
HLMO	Halamphora montana (Krasske) Levkov	31
HVEN	Halamphora veneta (Kützing) Levkov, var, veneta	59
KALA	Karayevia laterostrata (Hustedt) Bukhtiyarova	30
KCLE	Karayevia clevei (Grunow in Cl, et Grun,) Bukhtiyarova var, clevei	73
LGOE	Luticola goeppertiana (Bleisch in Rabenhorst)D,G, Mann in Round Crawford et Mann	98
LHUN	Lemnicola hungarica (Grunow) Round et Basson var, hungarica	35
LMUT	Luticola mutica (Kützing) D,G, Mann in Round Crawford et Mann var, mutica	39
LVCF	Luticola ventriconfusa Lange-Bertalot	47
MCIR	Meridion circulare (Greville) C,A, Agardh var, circulare	67
MPMI	Mayamaea permitis (Hustedt) Bruder et Medlin	30
MVAR	Melosira varians Agardh	51
NACI	Nitzschia acicularis Kützing) W,M,Smith var, acicularis	54
NAMP	Nitzschia amphibia f, amphibia Grunow var, amphibia	94
NANT	Navicula antonii Lange-Bertalot	96

Continued on next page

Table A.1 – *Continued*

Code	Taxa Name	Number of Images
NCAR	Navicula cari Ehrenberg var, cari	33
NCAT	Navicula catalanogermanica Lange-Bertalot et Hofmann	34
NCOM	Nitzschia communis Rabenhorst	30
NCPL	Nitzschia capitellata Hustedt in A, Schmidt et al, var, capitellata	67
NCPR	Navicula capitatoradiata Germain	66
NCRY	Navicula cryptocephala Kützing var, cryptocephala	91
NCTE	Navicula cryptotenella Lange-Bertalot var, cryptotenella	67
NCTO	Navicula cryptotenelloides Lange-Bertalot var, cryptotenelloides	53
NCTV	Navicula caterva Hohn et Hellerman	131
NDIS	Nitzschia dissipata subsp, dissipata (Kützing) Grunow var, dissipata	83
NERI	Navicula erifuga Lange-Bertalot in Krammer et Lange-Bertalot	68
NFIL	Nitzschia filiformis (W,M,Smith) Van Heurck var, filiformis	39
NFON	Nitzschia fonticola Grunow in Cleve et Möller var, fonticola	111
NGER	Navicula germainii Wallace	57
NGES	Nitzschia gessneri Hustedt	56
NGRE	Navicula gregaria Donkin var, gregaria	73
NHAN	Nitzschia hantzschiana Rabenhorst var, hantzschiana	49
NIAR	Nitzschia archibaldii Lange-Bertalot	36
NIBU	Nitzschia bulnheimiana (Rabenhorst) H,L,Smith var, bulnheimiana	57
NINC	Nitzschia inconspicua Grunow	43
NIPU	Nitzschia pusilla (Kützing) Grunow emend Lange-Bertalot	73
NLAN	Navicula lanceolata (Agardh) Ehrenberg var, lanceolata	35
NMIC	Nitzschia microcephala Grunow in Cleve et Moller var, microcephala	59

Continued on next page

Table A.1 – *Continued*

Code	Taxa Name	Number of Images
NPAE	Nitzschia paleacea (Grunow) Grunow in Van Heurck var, paleacea	125
NPAL	Nitzschia palea (Kützing) W,Smith var, palea	96
NRAD	Navicula radiosa Kützing var, radiosa	37
NRCH	Navicula reichardtiana Lange-Bertalot var, reichardtiana in LBK	119
NRCS	Navicula recens (Lange-Bertalot) Lange-Bertalot	37
NROS	Navicula rostellata Kützing var, rostellata	42
NSIA	Navicula simulata Manguin	30
NSOC	Nitzschia sociabilis Hustedt	92
NSTS	Nitzschia soratensis Morales et Vis	103
NSUA	Nitzschia subacicularis Hustedt in A, Schmidt et al,	67
NTPT	Navicula tripunctata (O,F,Müller) Bory var, tripunctata	48
NTRV	Navicula trivialis Lange-Bertalot var, trivialis	42
NULA	Nupela lapidosa (Krasske) Lange-Bertalot var, lapidosa	37
NVEN	Navicula veneta Kützing	92
NZAD	Nitzschia adamata Hustedt	35
NZSU	Nitzschia supralitorea Lange-Bertalot	81
PAPR	Parlibellus protractoides (Hustedt) Witkowski et Lange-Bertalot	37
PBIO	Psammothidium bioretii (Germain) Bukhtiyarova et Round	36
PDAO	Psammothidium daonense (Lange-Bertalot) Lange-Bertalot	38
PGRN	Planothidium granum (Hohn et Hellerman) Lange-Bertalot	32
PHEL	Psammothidium helveticum (Hustedt) Bukhtiyarova et Round var, helveticum	32
PLAU	Psammothidium lauenburgianum (Hustedt) Bukhtiyarova et Round	142
PLFR	Planothidium frequentissimum (Lange-Bertalot)Lange-Bertalot var, frequentissimum	144

Continued on next page

Table A.1 – *Continued*

Code	Taxa Name	Number of Images
PLHU	Platessa hustedtii (Krasske) Lange-Bertalot	36
PSAT	Psammothidium subatomoides (Hustedt) Bukhtiyarova et Round	35
PSBR	Pseudostaurosira brevistriata (Grun,in Van Heurck) Williams et Round var, brevistriata	93
PTCO	Platessa conspicua (A,Mayer) Lange-Bertalot	100
PTDE	Planothidium delicatulum (Kütz,) Round et Bukhtiyarova	55
PTDU	Planothidium dubium (Grunow) Round et Bukhtiyarova	50
PTLA	Planothidium lanceolatum (Brébisson ex Kützing) Lange-Bertalot var, lanceolatum	146
PULA	Punctastriata lancettula (Schumann) Hamilton et Siver	44
RABB	Rhoicosphenia abbreviata (C,Agardh) Lange-Bertalot	88
RSIN	Reimeria sinuata (Gregory) Kociolek et Stoermer	100
RUNI	Reimeria uniseriata Sala Guerrero et Ferrario	63
SANG	Surirella angusta Kützing var, angusta	34
SBKU	Surirella brebissonii var, kuetzingii Krammer et Lange-Bertalot	34
SBND	Staurosira binodis (Ehrenberg) Lange-Bertalot in Hofmann Werum et Lange-Bertalot	32
SCON	Staurosira construens Ehrenberg var, construens	63
SERA	Sellaphora radiosa (Hustedt) Kobayasi in Mayama et al,	31
SHTE	Stephanodiscus hantzschii f, tenuis (Hustedt) Håkansson et Stoermer	43
SIDE	Simonsenia delognei Lange-Bertalot	87
SPUP	Sellaphora pupula (Kützing) Mereschkowksy var, pupula	90
SSEM	Sellaphora seminulum (Grunow) D,G, Mann	65
SSVE	Staurosira venter (Ehrenberg) Cleve et Moeller var, venter	82
SVTL	Sellaphora ventraloides (Hustedt) Falasco et Ector	49
TAPI	Tryblionella apiculata Gregory	48

Continued on next page

Table A.1 – *Continued*

Code	**Taxa Name**	**Number of Images**
TFAS	Tabularia fasciculata (Agardh) Williams et Round	67
UULN	Ulnaria ulna (Nitzsch) Compère var, ulna	34

Appendix B

Morphological Parameters Formulas

Table B.1: List of morphological parameters extracted from a diatom's contour.

Parameter	**Definition/Formula**
Length (l)	Largest dimension of the masks' bounding box.
Width (w)	Smallest dimension of the masks' bounding box.
Area (A)	The space covered by the mask region.
Perimeter (P)	The length of the mask's contour.
Ellipticity	$\frac{A}{\text{Area of ellipse that fits the mask}}$
Eccentricity	$\sqrt{1 - \frac{w^2}{l^2}}$
Roundness	$\frac{4\pi A}{P^2}$
Convexity	$\frac{A}{\text{Area of convex hull of the mask}}$
Concavity	1 - convexity
`length-width-ratio`	$\frac{l}{w}$
`area-perimeter-ratio`	$\frac{A}{P}$
`area-perimeter-sq`	$\frac{A}{P^2}$
Rectangularity	$\frac{A}{\text{Area of minimum bounding box enclosing the mask}}$
Hu Moments [75]	Seven statistical features used to capture different properties of the image such as its centroid, area, and orientation

Table B.2: List of parameters extracted from the Gray-Level Co-occurrence Matrix (GLCM). G is the co-occurence matrix, i and j are the location in the matrix, and n is the size of the co-occurence matrix, $\mu_x = \sum_{i,j=0}^{n-1} i(G(i,j))$, $\mu_y = \sum_{i,j=0}^{n-1} j(G(i,j))$, $\sigma_x = \sqrt{\sum_{i,j=0}^{n-1} G(i,j)(i-\mu_x)^2}$ and $\sigma_y = \sqrt{\sum_{i,j=0}^{n-1} G(i,j)(j-\mu_y)^2}$

Parameter	**Definition/Formula**
Contrast	$\sum_{i,j=0}^{n-1} G(i,j)(i-j)^2$
Dissimilarity	$\sum_{i,j=0}^{n-1} G(i,j)\|i-j\|$
Homogeneity	$\sum_{i,j=0}^{n-1} j\frac{G(i,j)}{1+(i-j)^2}$
Correlation	$\sum_{i,j=0}^{n-1} \frac{G(i,j)[(i-\mu_x)(j-\mu_y)]}{\sigma_x\sigma_y}$
Angular Second Moment (ASM)	$\sum_{i,j=0}^{n-1} G(i,j)^2$
Energy	$\sqrt{ASM}$

Table B.3: First order statistical parameters extracted from masked diatom image and Local Binary Pattern (LBP). Let the total number of pixels in the image be N and let $p(i)$ be the pixel value at location i.

Parameter	**Definition/Formula**
Mean (μ)	$\frac{1}{N}\sum_{i=0}^{N-1} p(i)$
Standard deviation (σ)	$\sqrt{\frac{1}{N}\sum_{i=0}^{N-1}(p(i)-\mu)^2}$
Variance	σ^2
Entropy	$-\sum_{i=0}^{N-1} p(i)\log_2(p(i))$
Asymmetry	$A = \frac{1}{N}\sum_{i=0}^{N-1}\|p(i)-\mu\|$
Skewness	$\frac{\frac{1}{N}\sum_{i=0}^{N-1}(p(i)-\mu)^3}{\sigma^3}$
Kurtosis	$\frac{\frac{1}{N}\sum_{i=0}^{N-1}(p(i)-\mu)^4}{\sigma^4}$
First quartile Q_1	First quartile (25^{th} percentile) of pixel values
Second quartile Q_2	Second quartile (50^{th} percentile) of pixel values
Third quartile Q_3	Third quartile (75^{th} percentile) of pixel values
Interquartile Range (IQR)	$Q_3 - Q_1$

Appendix C

Supplementary Materials for Chapter 5

Table C.1: List of taxa used for training the classifier

Code	Misclassified Taxa	Misclassification Rate (%)
AAMB	AUDI	3.45
	AUGR	6.90
	CAGR	6.90
ACAF	ADKR	7.14
ACLI	ADEU	2.70
	ADMI	5.41
	EOMI	2.70
ACOF	HACO	66.67
ADCT	ADMI	7.14
	ADSB	2.38
ADEU	ACAF	2.13
	ADMI	4.26
	ADSB	2.13
ADKR	EORH	8.33
ADLA	ADPY	8.33
ADMI	ACLI	2.50
	ADGL	1.25

Continued on next page

Table C.1 – *Continued*

Code	Misclassified Taxa	Misclassification Rate (%)
	ADSA	2.50
ADPY	ENCM	2.78
ADSA	ADEU	7.14
	ADLA	7.14
	ADMI	7.14
ADSH	ADEU	13.33
	PSBR	6.67
ADSU	ADPY	31.25
	ADSB	6.25
	GMPU	6.25
	NFON	12.50
AOVA	ACOP	8.33
	CPLA	8.33
AUDI	AUSU	75.00
AUGA	AUGR	20.00
	CMLF	6.67
AUSU	AUDI	58.33
	CDUB	8.33
BNEO	CTUM	14.29
CAGR	DPST	3.23
CATO	CMED	11.11
CBKU	LGOE	14.29
CDUB	CDTG	3.33
	CMED	3.33
	CMEN	3.33
CEUG	CPED	3.85
	DOCU	3.85
CINV	CAGR	5.88
CMED	CINV	3.85
CNDI	COPL	11.11
COCE	CPOL	5.26
COPL	CEUG	12.00

Continued on next page

Table C.1 – *Continued*

Code	Misclassified Taxa	Misclassification Rate (%)
CPED	CEUG	7.69
DMES	MPMI	9.09
DMON	DITE	7.69
	DMES	7.69
DSTE	CMED	7.14
ECPM	ENCM	7.14
EEXI	ETEN	14.29
EMIN	FSBH	11.11
ENMI	ESLE	5.88
	GBOB	5.88
ENVE	ESLE	3.33
EOCO	GELG	3.70
EOMI	ADEU	5.56
	PSAT	5.56
FCRO	FGRA	10.00
FGRA	FVAU	4.35
FSAP	MPMI	5.56
GACC	APED	5.56
GCLF	GMPU	10.00
GELG	PSBR	10.00
GLAT	GELG	20.00
	GPRI	10.00
	LGOE	10.00
GMIC	GPRO	7.69
GOLI	GOLD	5.88
GPEL	GELG	11.11
GPRI	GBOB	4.17
	GELG	8.33
	GPLI	4.17
GTER	GMIN	9.52
	PTDU	4.76
HACO	ACOF	100.00

Continued on next page

Table C.1 – *Continued*

Code	Misclassified Taxa	Misclassification Rate (%)
HLMO	ADMI	16.67
MCIR	BNEO	7.69
	GPLI	7.69
MPMI	ACAF	16.67
NANT	DSTE	5.26
NCPL	NFON	7.69
	NPAL	15.38
NCRY	NCTV	5.56
NDIS	PTCO	5.88
NERI	NCRY	7.14
NGES	FAUT	9.09
	NPAE	9.09
	TFAS	9.09
NHAN	NFON	10.00
NINC	NAMP	11.11
NPAE	NIPU	4.00
NRCH	GELG	4.17
NSTS	DMON	4.76
NTPT	NLAN	10.00
	PSBR	10.00
NZAD	NCPL	14.29
NZSU	NPAE	6.25
PAPR	GCBC	14.29
PLAU	ADSB	6.90
PLFR	AFUG	3.45
	CNTH	3.45
	GTER	3.45
	PTCO	3.45
	PTLA	6.90
PLHU	GPSA	14.29
PSAT	PDAO	14.29
PTCO	PTDU	5.00
PTLA	GPSA	3.45

Continued on next page

Table C.1 – *Continued*

Code	Misclassified Taxa	Misclassification Rate (%)
	PLFR	3.45
RABB	GPLI	16.67
RSIN	FVAU	5.00
	GELG	5.00
	RUNI	5.00
RUNI	CAEX	7.69
SCON	AUPU	7.69
	SSVE	7.69
SERA	ECAE	16.67
SIDE	EOCO	5.88
TFAS	PSBR	7.69
UULN	FVAU	14.29

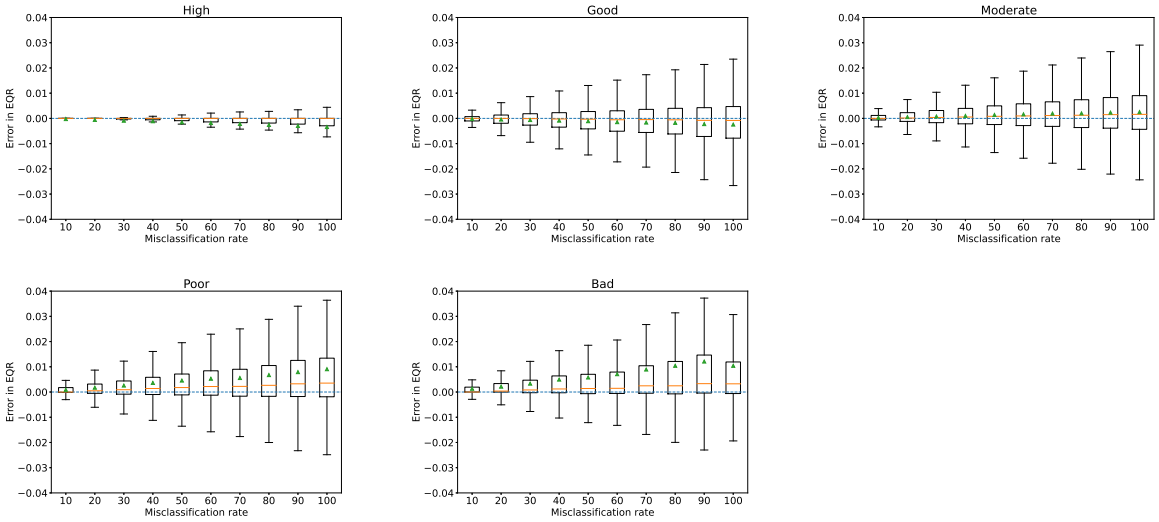


Figure C.1: Box plot of the EQR errors (difference between the EQR scores obtained before and after introducing misclassification) for the 5 ecological status *when misclassification is introduced using a CNN classifier*. The misclassification rates are from 10% to 100%. The orange line in each box plot represents the median value, and the green point is the mean error. The box represents the interquartile range (IQR), the whiskers represent the minimum and maximum range in the data.

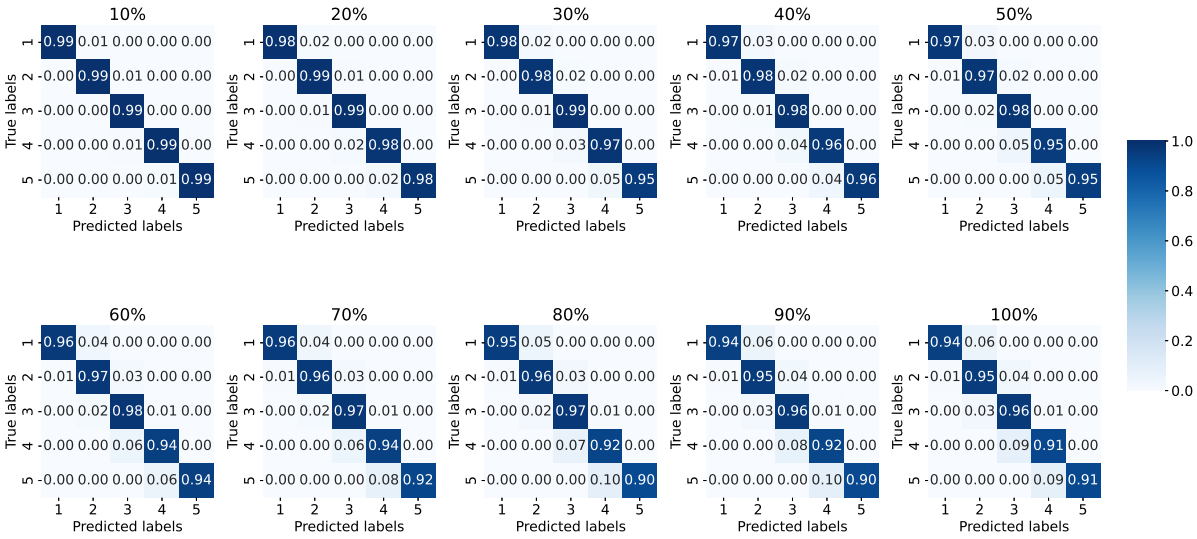


Figure C.2: Confusion matrices for the 5 ecological status for different *misclassification values introduced using a CNN classifier*. (Top row) Misclassification level from 10 to 50% from left to right. (Bottom row) Misclassification level from 60 to 100% from left to right. The class numbers 1,2,3,4 and 5 corresponds to the classes 'High', 'Good', 'Moderate', 'Poor' and 'Bad' respectively. The X-axis is the predicted label and the Y-axis is the ground truth labels. The values in the matrices are the % of inventories that belong to a particular status in the ground truth labels and were predicted as another status when introducing misclassification.

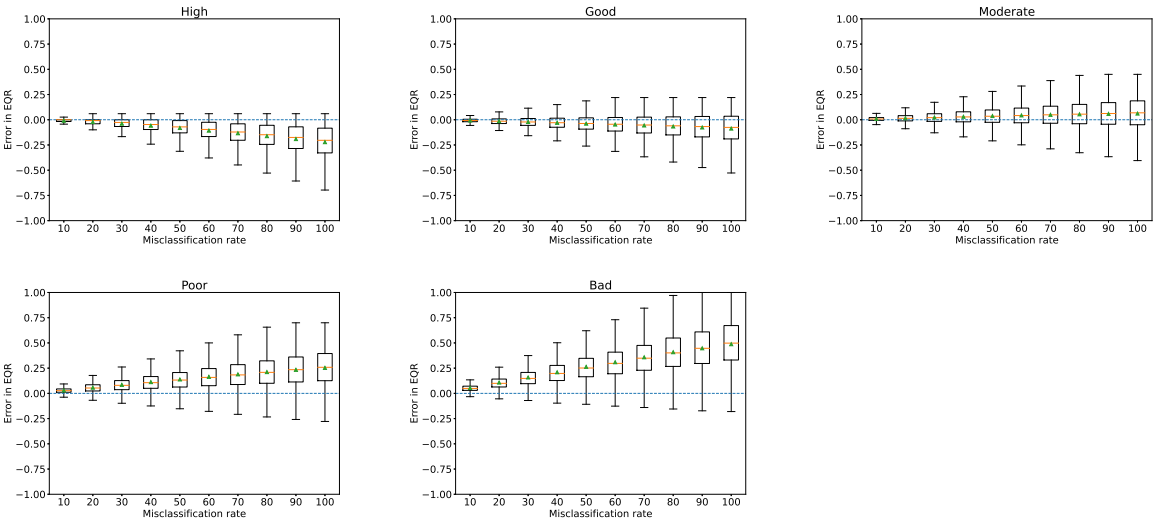


Figure C.3: Box plot of the EQR errors (difference between the EQR scores obtained before and after introducing misclassification) for the 5 ecological status *when misclassification is introduced randomly*. The misclassification rates are from 10% to 100%. The orange line in each box plot represents the median value, and the green point is the mean error. The box represents the interquartile range (IQR), the whiskers represent the minimum and maximum range in the data.

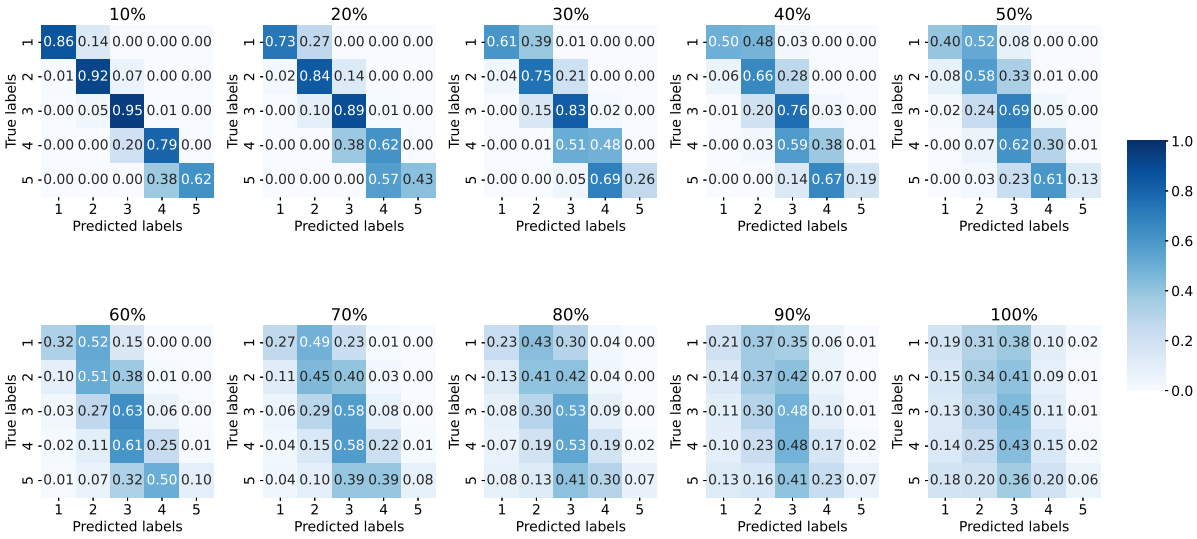


Figure C.4: Confusion matrices for the 5 ecological status for different *misclassification values introduced randomly*. (Top row) Misclassification level from 10 to 50% from left to right. (Bottom row) Misclassification level from 60 to 100% from left to right. The class numbers 1,2,3,4 and 5 corresponds to the classes 'High', 'Good', 'Moderate', 'Poor' and 'Bad' respectively. The X-axis is the predicted label and the Y-axis is the ground truth labels. The values in the matrices are the % of inventories that belong to a particular status in the ground truth labels and were predicted as another status when introducing misclassification.

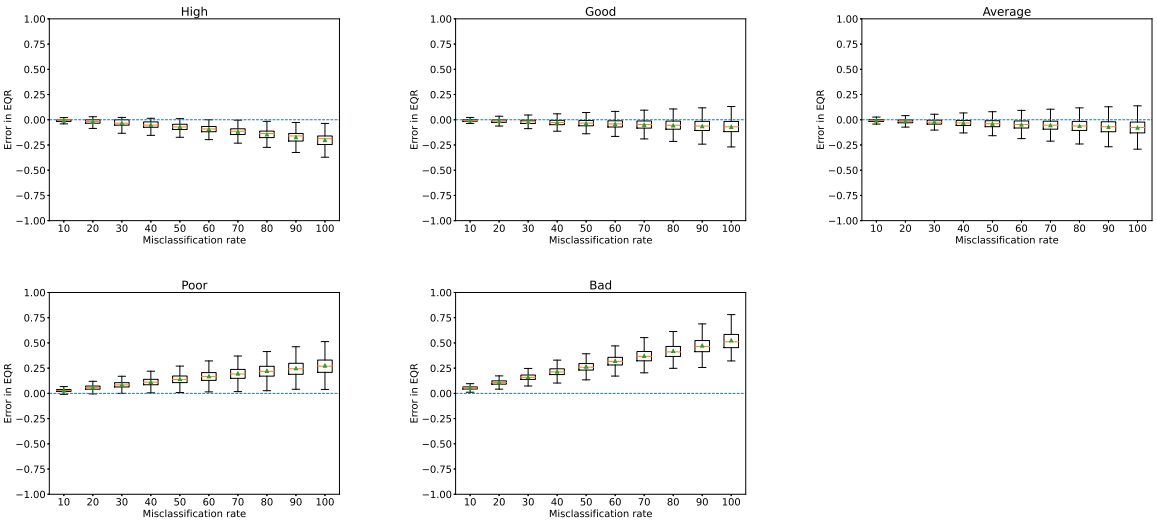


Figure C.5: Box plot of the EQR errors (difference between the EQR scores obtained before and after introducing misclassification) for the 5 ecological status *when misclassification is introduced at the genus level of the diatom hierarchy*. The misclassification rates are from 10% to 100%. The orange line in each box plot represents the median value and the green point is the mean error. The box represents the interquartile range (IQR), the whiskers represent the minimum and maximum range in the data.

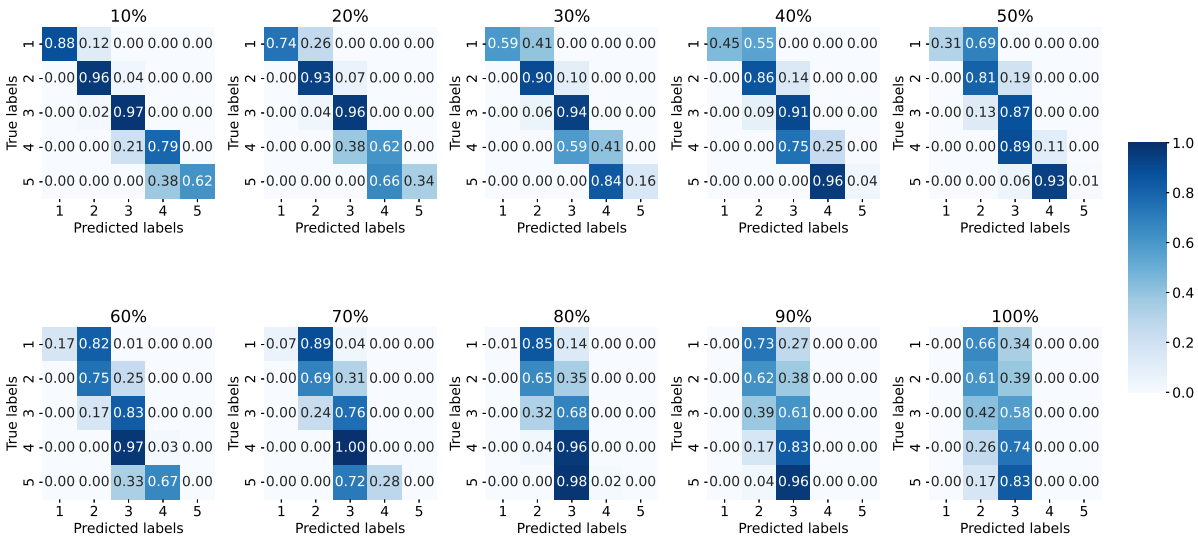


Figure C.6: Confusion matrices for the 5 ecological status for different *misclassification values introduced at the genus level of the diatom hierarchy.* (Top row) Misclassification level from 10 to 50% from left to right. (Bottom row) Misclassification level from 60 to 100% from left to right. The class numbers 1,2,3,4 and 5 corresponds to the classes 'High', 'Good', 'Moderate', 'Poor' and 'Bad' respectively. The X-axis is the predicted label and the Y-axis is the ground truth labels. The values in the matrices are the % of inventories that belong to a particular status in the ground truth labels and were predicted as another status when introducing misclassification.