



**HAL**  
open science

# A nearest-neighbours kernel for classification: a case study of in situ two-dimensional plankton images with correction of total volume estimates for copepods

Cédric Dubois

## ► To cite this version:

Cédric Dubois. A nearest-neighbours kernel for classification: a case study of in situ two-dimensional plankton images with correction of total volume estimates for copepods. Modeling and Simulation. Université Côte d'Azur, 2023. English. NNT: 2023COAZ4032 . tel-04264556

**HAL Id: tel-04264556**

**<https://theses.hal.science/tel-04264556>**

Submitted on 30 Oct 2023

**HAL** is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

# THÈSE DE DOCTORAT

Un noyau des plus proches voisins  
pour la classification : application aux  
images de plancton bidimensionnelles  
*in situ* avec correction des estimations  
de volume total pour les copépodes

**Cédric DUBOIS**

Équipe MORPHEME – Inria Sophia Antipolis/I3S/iBV

**Présentée en vue de l'obtention  
du grade de docteur en** Automatique, traite-  
ment du signal et des images  
d'Université Côte d'Azur

**Dirigée par :** Eric DEBREUVE, Chargé de  
recherche, Université Côte d'Azur

**Co-encadrée par :** Jean-Olivier IRISSON,  
Maître de conférences, Sorbonne Université

**Soutenue le :** 30 mars 2023

**Devant le jury, composé de :**

**Rapporteurs :**

Émilie POISSON CAILLAULT, Maî-  
tresse de conférences, Université du Littoral  
Côte d'Opale

Ketil MALDE, Full professor, University  
Of Bergen

Bertrand GRANADO, Professeur des uni-  
versités, Sorbonne Université

**Examineurs :**

Frédéric PRECIOSO, Professeur des uni-  
versités, Université Côte d'Azur

Frédéric MAPS, Full professor, Université  
Laval

**Invités :**

Jean-Olivier IRISSON, Maître de confé-  
rences, Sorbonne Université

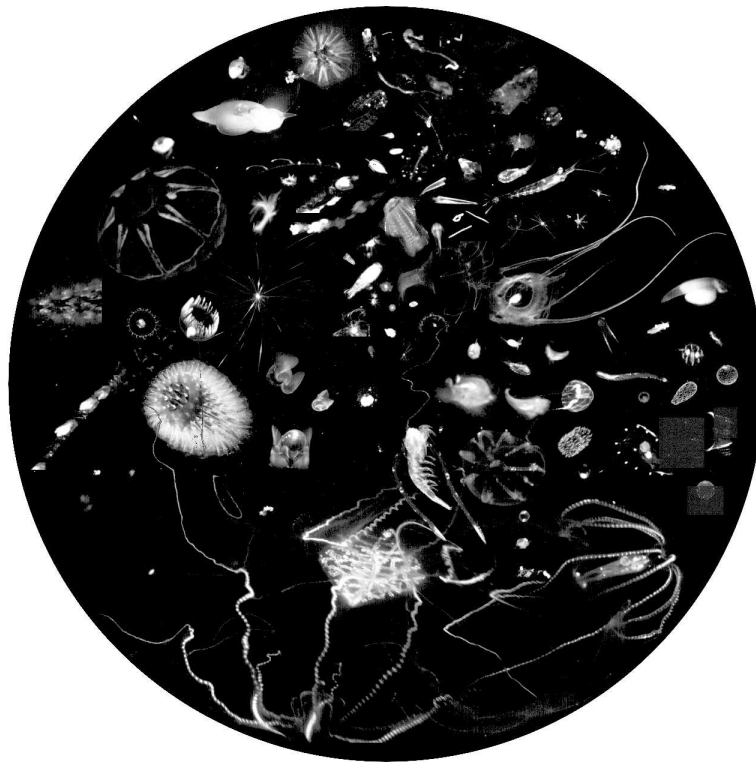


A Nearest-Neighbours Kernel for Classification: a Case Study of *In Situ* Two-Dimensional Plankton Images with Correction of Total Volume Estimates for Copepods

Cédric Dubois

Under the supervision of Éric Debreuve and co-supervision of Jean-Olivier Irisson

30 march 2023







## Abstract

Plankton organisms are a key component of the biosphere: they are at the base of marine food webs and are important contributors to biogeochemical cycles, notably of carbon, nitrogen and oxygen. Indeed, phytoplankton captures carbon dioxide from the atmosphere and produces dioxygen; zooplankton contributes to aggregate and export this carbon at depth, where it is sequestered for hundreds of years. This so-called ‘biological carbon pump’ is studied by ecologists to estimate its efficiency nowadays and in the future, in response to climate change. A modern approach consists in studying how the environment is linked with the functioning of ecosystems through ‘traits’ (*i.e.*, individual characteristics) of organisms. For example, a high correlation has been observed between the size distribution of zooplankters and the carbon sequestration efficiency. *In situ* imaging instruments and large image databases have been built for plankton, allowing taxonomic classification of organisms and quantification of the total volume of each group based on their morphology. The development of automated classification methods has been essential to help ecologists process data. Among them, Artificial Neural Networks (ANNs) have proven to be efficient and accurate, but their decisions are often hard to interpret. On one hand, in this thesis, we put forward the idea that following the transform-then-classify-simply approach of ANNs using a simple, explicit, transform can result in a classifier whose predictions are both interpretable (thus, trustable) and accurate. The proposed transform is defined as a linear combination of per-class targets, and the classification is performed, like with ANNs, by a nearest-target decision. Furthermore, as a main theoretical result, we establish that the proposed transform with equidistant targets defines a kernel associated with the Weighed-k-Nearest-Neighbor (W-kNN) classifier, and allows interpreting the W-kNN classifier as a member of a larger family of target-based classifiers, which satisfies an optimality criterion. We propose a modern W-kNN implementation of high enough computational efficiency to deal with large datasets, like the ones collected every day by plankton imaging instruments. We were therefore able to perform a leave-one-out cross-validation on large plankton images datasets. On another hand, we tackle the correction of the estimation of copepods volume from two-dimensional *in situ* images. Copepods are the most abundant zooplankton group and represent a significant share of the biomass of animals on Earth. The standard volume estimation methods are biased due to the effect of the projection onto the image plane. Two such methods exist: based on the Equivalent Spherical Diameter (ESD) and based on extending the best-fitting ellipse to 3D. We present a procedure for correcting the total volume estimations of both methods for this zooplankton group. First, the projection of the body of the copepod is robustly extracted. Second, we note that the exact projection of an ellipsoidal body model onto the image plane is an ellipse. Therefore, based on the simulation of many realistic ellipsoids (relying on shape distributions established from manual size measurements on a dataset) and their projections from random point of views, we can compute a total volume correction factor for each standard method.

As opposed to a new volume estimation method from the images, the proposed correction factors allow improving the estimations of past studies, while being applicable to future studies as well. To validate the proposed method, we applied it to a database of 150,000 images of copepods captured by the UVP, and found that the corrections decreased the gap between the two standard methods by a factor of 50. The correction factors indicated that the ESD method tends to over-estimate the total volume by around 20% and the ellipse method under-estimates it by around 10%.

**Keywords:** kernel, nearest-neighbours, classification, plankton, zooplankton, bio-volume, copepod, image, *in situ*

## Résumé en français

Les organismes qui composent le plancton sont des éléments essentiels de la biosphère : à la base de la chaîne alimentaire marine, ils sont au cœur des cycles biogéochimiques, notamment du carbone, de l'azote et de l'oxygène. En effet, le phytoplancton capte le dioxyde de carbone de l'atmosphère et produit du dioxygène ; le zooplancton contribue à exporter ce carbone en profondeur. Les écologues étudient cette « pompe à carbone biologique », afin d'évaluer son efficacité actuelle et future face au changement climatique. Une approche moderne consiste à étudier la manière dont l'environnement est lié au fonctionnement des écosystèmes par le biais des « traits » (caractéristiques individuelles) des organismes. Une corrélation importante a été observée entre la distribution des tailles des zooplanctons et l'efficacité de la séquestration du carbone. Des instruments d'imagerie *in situ* et de grands jeux de données d'images ont été mis en œuvre pour le plancton, permettant la classification taxonomique des organismes et la quantification du volume total par groupe. Le développement de méthodes de classification automatisée a été essentiel pour l'assistance au traitement des données. À ce titre, les Réseaux de Neurones Artificiels (RNAs) se sont avérés très utiles et précis, mais leurs décisions sont souvent difficiles à interpréter. Dans un premier temps, nous montrons que l'approche transformation-puis-classification-simple des RNAs avec une transformation simple et explicite, conduit à une méthode de classification dont les prédictions sont interprétables (donc fiables) et précises. La transformation proposée est définie comme une combinaison linéaire de cibles par classe. Ensuite, la classification est effectuée, comme avec les RNAs, en prenant la cible la plus proche. Notre résultat principal démontre que, pour des cibles équidistantes, cette transformation définit un noyau associé au classifieur des k-plus-Proches-Voisins-Pondérés (kPPP). Ceci permet d'interpréter les kPPP comme un membre d'une famille plus large de classifieurs utilisant des cibles. Nous proposons une implémentation moderne des kPPP suffisamment efficace pour traiter de grands ensembles de données, tels que ceux collectés chaque jour par les instruments d'imagerie du plancton. Nous avons ainsi effectué une validation croisée avec l'omission d'un échantillon sur de grands jeux de données d'images de plancton. Dans un second temps, nous étudions l'estimation du volume des copépodes à partir d'images bidimensionnelles *in situ*. Les copépodes constituent le groupe zooplanctonique le plus abondant. Les deux méthodes standards d'estimation du volume sont biaisées en raison de l'effet de la projection sur le plan de l'image. L'une utilise le Diamètre Équivalent Sphérique (DES) et l'autre, l'ajustement d'une ellipse. Nous présentons une procédure pour corriger les estimations de volume total des deux méthodes pour ce groupe. La projection du corps du copépode seulement est extraite. Nous observons en outre que la projection exacte d'une ellipsoïde sur le plan est une ellipse. Par conséquent, à partir de la simulation de nombreuses ellipsoïdes réalistes (grâce à des mesures de taille manuelles) et de leurs projections selon une orientation aléatoire, nous calculons un facteur de correction du volume total par méthode. Contrairement à une nouvelle

méthode d'estimation, les corrections proposées permettent d'améliorer les estimations des études passées, tout en étant applicables aux prochaines. À titre de validation, nous appliquons la procédure de correction aux estimations du volume total de 150 000 copépodes à partir d'images prises par un instrument *in situ*. Les facteurs corrections permettent de réduire l'écart entre les deux estimations d'un facteur 50, et indiquent que la méthode DES tend à surestimer le volume total d'environ 20 % et que celle utilisant l'ellipse tend à le sous-estimer d'environ 10 %.

**Mots clés :** noyau, plus-proches-voisins, classification, plancton, zooplancton, bio-volume, copépode, image, *in situ*

## Remerciements

Je remercie le personnel du laboratoire I3S, de l'UCA et de l'Inria pour avoir maintenu un cadre très agréable à vivre. J'ai grandement apprécié venir au laboratoire avec enthousiasme. Plus particulièrement, je tiens à remercier Frédéric, Nadia et Pierre de l'I3S pour leur accompagnement administratif, leur réactivité, leur bienveillance et leur gentillesse.

Je remercie chaleureusement tous les membres permanents de l'équipe Morpheme, Éric, Grégoire, Laure, Luca et Xavier pour les nombreux moments conviviaux et les échanges constructifs que l'on a partagés. Plus généralement, je remercie toutes les personnes qui ont été ou sont toujours présentes dans l'équipe pour leurs conseils, leur écoute et leur bonne humeur quotidienne. Je remercie Éric de m'avoir guidé dans ce parcours doctoral, de m'avoir appris, montré, partagé et expliqué tant de choses pendant ces années, des principes fondamentaux en sciences à la mécanique de vélo. En particulier, je suis très reconnaissant envers lui pour tous ces moments de discussions qui m'ont permis d'évoluer, de grandir et de mener à bien l'exercice du doctorat. Je remercie Jean-Olivier de m'avoir amené à des sujets de recherche qui m'étaient alors inconnus, et de son investissement pendant ces années. Je le remercie d'avoir pris le temps de m'expliquer pédagogiquement des concepts en écologie et les expériences menées par le LOV et ailleurs, et d'avoir répondu à mes nombreux questionnements. Je remercie les membres de l'équipe COMPLEX du LOV de m'avoir accueilli et présenté leurs travaux. Plus particulièrement, je remercie Fabien, Laetitia, Marc et Thelma pour les échanges constructifs que nous avons eus.

Je remercie grandement Émilie Poisson Caillault, Ketil Malde et Bertrand Granado d'avoir relu ce manuscrit. Je les remercie d'avoir apporté un regard extérieur sur mon travail accompagné de remarques pertinentes qui ont amélioré la qualité de ce document. Je remercie les examinateurs de la soutenance Frédéric Precioso et Frédéric Maps pour m'avoir écouté attentivement présenter mes travaux. Je remercie plus généralement tous les membres du jury, qui ont instauré un climat professionnel et bienveillant lors de la soutenance, et qui ont mené une séance de questions et de discussions pertinentes et constructives.

Je remercie mon camarade de café et de mécanique Bastien (a.k.a the O.G. expert, a.k.a. Jordan) pour avoir entretenu notre lieu de travail comme il se doit, pour avoir supporté mes boutades et autres blagues d'altitudes qui se mesurent en centimètres ; ses conseils 'Top Gear' en optimisation de turbines et son répondant d'ingénieur de la NASA lors de nos débats, intéressants. Je le remercie pour la conception et le partage de ses '*templates*' prestigieux de bonne facture. Je remercie Marie Guyomard pour avoir rétabli un niveau convenable quand il était trop aérien, et d'être allé toujours plus loin, trop loin. Je la remercie de m'avoir motivé à aller à la piscine et de m'avoir appris à nager correctement par la même occasion. Je la remercie pour son soutien précieux et ses mots pour rire de toutes les situations. Je remercie mon fidèle graphiste Baptiste Schall (suivez @baptisteschall) de m'avoir prêté ses talents d'artiste peintre. Plus particulièrement, pour avoir grandement participé à l'élaboration des Figures 1 et 2 de

l'introduction de ce document, avec, notamment, la modélisation 3d réaliste d'organismes planctoniques. Je le remercie également pour avoir été mon fidèle compagnon d'escalade de l'extrême. Je remercie l'écurie Bastah Motorsport – FC chômage pour la confiance qu'elle a placée en mes talents. Je remercie tous ses membres pour cette merveilleuse aventure ainsi que mon unique sponsor MAXSexcul pour m'avoir fourni leur unique prototype de calculateur Plug & Pay pour Mégane 2 dci. Merci l'ekip.

Je remercie mes collègues de l'université d'Aix-Marseille de m'avoir accompagné durant toutes ces années folles, et qui m'ont permis d'arriver jusqu'à ce doctorat. Je remercie chaleureusement Adrien, Lorenzo, Romain et Valentin pour leur amitié.

Je remercie Adeline Marcellino, pour m'avoir accompagné, encouragé et fait évoluer, fidèlement depuis le début de cette aventure de thèse qui concorde avec notre belle aventure.

Je suis particulièrement reconnaissant envers les membres de ma famille, qui m'ont accompagné jusqu'ici, et m'ont encouragé et permis d'accomplir mes études.

Merci à toutes, merci à tous.



*À mamie Colette,*



## Glossary

- CV<sub>LOO</sub>** Leave-One-Out Cross-Validation 9, 16, 33, 35–39, 45, 46, 48, 50
- k*-NN** *k*-Nearest-Neighbours 16, 36, 37, 39, 116, 117
- ACC** Accuracy 15, 36–39, 42, 45, 116, 117
- ANN** Artificial Neural Network 7, 8, 13, 16, 17, 99
- B-ACC** Balanced Accuracy 42, 45, 46, 48, 50, 52
- CNN** Convolutional Neural Network 5, 8, 17, 38, 41, 45, 46, 48, 50, 52, 99, 118–120
- CV** Cross-Validation 16, 31, 35–39, 42, 45, 46, 48
- DR** Dimension Reduction 31
- ERM** Empirical Risk Minimization 15
- ESD** Equivalent Spherical Diameter 3, 5, 97
- GPU** Graphics Processing Unit 33, 34, 36, 37, 39, 46
- KDE** Kernel Density Estimation 80–82, 85
- LOV** Laboratoire d’Océanographie de Villefranche 43, 45, 46, 56
- NBSS** Normalized Bio-volume Size Spectra 93–97, 100
- NN** Nearest-Neighbour 9
- NN-**Kernel**** Nearest-Neighbour-Kernel xvi, 34, 36, 38
- NT** Nearest Target 17, 27, 35, 36, 38, 99
- PCA** Principal Component Analysis 39, 46, 48
- RF** Random Forest 7, 36, 37, 41, 45–48, 50, 51, 100

**SVM** Support Vector Machine 7, 16, 17, 36–38, 50, 100

**t-SNE** t-Stochastic Neighbor Embedding 31, 115

**UVP** Underwater Vision Profiler 5

**UVP5** Underwater Vision Profiler 5 3, 5, 47, 56–59, 89, 95, 100

**UVP5-Cop** Underwater Vision Profiler 5 *Copepod* xvi, xvii, 56, 57, 61, 66, 68, 81, 83, 84, 87–90, 94, 95

**UVP5-HD** Underwater Vision Profiler 5 - High Definition 47–49

**W-*k*-NN** Weighed-*k*-Nearest-Neighbours 33–39, 41, 42, 45, 46, 48, 50–52, 116–121

**WNN** Weighed-Nearest-Neighbours 9, 16, 19, 27, 31, 34, 99, 100

# Contents

<b>Abstract</b>	<b>v</b>
<b>Résumé</b>	<b>vii</b>
<b>Acknowledgements</b>	<b>ix</b>
<b>Glossary</b>	<b>xiii</b>
<b>Contents</b>	<b>xv</b>
<b>1 Introduction</b>	<b>1</b>
1.1 Definitions . . . . .	2
1.2 Context . . . . .	2
1.2.1 Plankton Ecology . . . . .	2
1.2.2 Biogeochemical fluxes at the ocean surface . . . . .	4
1.3 Images of plankton . . . . .	5
1.3.1 EcoTaxa: processing millions of plankton images . . . . .	5
1.3.2 Classifying plankton images . . . . .	5
1.4 Contributions . . . . .	9
<b>I A Nearest-Neighbours Kernel for Classification: a case study of Plankton Images</b>	<b>11</b>
<b>2 A Kernel Associated to the Nearest-Neighbours</b>	<b>13</b>
2.1 Reminders on supervised classification . . . . .	14
2.1.1 Empirical Risk Minimization . . . . .	14
2.1.2 Evaluating the generalization performances . . . . .	15
2.2 Motivations . . . . .	16
2.3 Proposed Classifier . . . . .	17
2.3.1 Definition . . . . .	17
2.3.2 Additional Notations . . . . .	18
2.3.3 Properties . . . . .	19
2.4 Influence of Target Positions . . . . .	20
2.4.1 Problem Statement . . . . .	20
2.4.2 Case 1/3: $\hat{k} \neq m$ and $l \neq m$ (applies only for $p \geq 3$ ) . . . . .	21
2.4.3 Case 2/3: $\hat{k} = m$ (consequently, $l \neq m$ ) . . . . .	22

2.4.4	Case 3/3: $l = m$ (consequently, $\hat{k} \neq m$ )	23
2.5	Optimal Targets for 3 Classes or More	24
2.5.1	Variational Formulation	24
2.5.2	Minimization of $F$	25
2.5.3	Classification Point-of-View	27
2.6	Selection of $\gamma$	28
2.6.1	Limit Cases	28
2.6.2	About an Optimal Value	30
2.7	Some Choices of parameters $\gamma_i, i \in [1..n]$	31
<b>3</b>	<b>Implementation and Experimental Results</b>	<b>33</b>
3.1	Purpose	34
3.2	The method in practice	34
3.2.1	Asymptotically optimal weights	34
3.2.2	Leave-One-Out Cross-Validation	35
3.2.3	Per-class weights for unbalanced data sets	35
3.2.4	Numerical implementation with KeOps	36
3.3	Experimental results on synthetic data	36
3.3.1	Comparison to standard methods	36
3.3.2	Procedure to use the NN-Kernel	38
3.4	Experimental results on CIFAR-10	38
<b>4</b>	<b>Experimental Results on Plankton Images</b>	<b>41</b>
4.1	Introduction	42
4.2	ZooScan instrument & ZooScanNet data set	42
4.2.1	Presentation	42
4.2.2	ZooProcess: Handcrafted Features	42
4.2.3	ZooScan image features from fine-tuned CNN	46
4.3	UVP5-HD instrument & data set	47
4.3.1	UVP5-HD <i>in situ</i> imaging instrument	47
4.3.2	UVP5-HD image classification with features from a fine-tuned CNN	48
4.3.3	UVP5-HD Copepods	50
<b>II</b>	<b>Correcting the Estimations of Copepod's Total Volume from 2-d <i>In Situ</i> Imaging</b>	<b>53</b>
<b>5</b>	<b>Copepods' Bio-Volume Estimates from 2-d <i>In Situ</i> Images</b>	<b>55</b>
5.1	Introduction	56
5.2	Imaging the copepods worldwide: the UVP5-Cop dataset	57
5.3	Standard geometrical measurements with ZooProcess	57
5.4	Standard methods for volume estimations	59
5.4.1	Optical model	59
5.4.2	Using the equivalent spherical diameter ( $\mathcal{M}_{\text{ESD}}$ )	60
5.4.3	Using a best-fitting ellipse ( $\mathcal{M}_{\text{ELL}}$ )	60
5.5	Limits of the Standard Methods	61
5.5.1	Illustration of the limits of the current method	61
5.5.2	Illustration of $\mathcal{M}_{\text{ESD}}$ & $\mathcal{M}_{\text{ELL}}$ error	61
5.5.3	Discrepancy between total volume estimations	61

<b>6</b>	<b>Extracting the Copepod Prosome from 2-d Images</b>	<b>65</b>
6.1	Motivations : influence of antennas . . . . .	66
6.2	Geometrical measurements . . . . .	66
6.2.1	Common process . . . . .	66
6.2.2	Area estimation and ellipse fitting . . . . .	67
6.3	Application to UVP5-Cop images . . . . .	68
6.4	Discussion . . . . .	68
<b>7</b>	<b>Modelling the projection of a copepod's prosome</b>	<b>73</b>
7.1	Introduction . . . . .	74
7.2	Projection of an ellipsoid . . . . .	74
7.2.1	Geometrical setup . . . . .	74
7.2.2	Ellipsoid silhouette in 3-d . . . . .	75
7.2.3	2-d silhouette . . . . .	76
7.2.4	Semi-axes for parallel projection . . . . .	77
7.3	Volume estimation errors . . . . .	77
7.3.1	Expressions for the standard methods . . . . .	77
7.3.2	Invariance of errors to scaling . . . . .	78
7.4	Simulation of realistic ellipsoids . . . . .	79
7.4.1	A simulation for the total volume correction factors . . . . .	79
7.4.2	Estimating shape parameters . . . . .	81
7.4.3	Distribution of individual errors . . . . .	81
<b>8</b>	<b>Correcting Total Copepods' Volume Estimates</b>	<b>83</b>
8.1	Total volume correction . . . . .	84
8.1.1	Proposed approach . . . . .	84
8.1.2	Invariance of total volume estimation error to size normalization . . . . .	85
8.2	Experimental results with the UVP5-Cop dataset . . . . .	87
8.3	Robustness of the method . . . . .	88
8.3.1	Shape . . . . .	88
8.3.2	Orientation . . . . .	89
8.3.3	Number of simulated ellipsoids . . . . .	90
8.4	Discussion . . . . .	90
<b>9</b>	<b>Insight on the error on Normalized Bio-volume Size Spectra</b>	<b>93</b>
9.1	Introduction . . . . .	94
9.2	Method for estimating the major semi-axes $r_1$ distribution . . . . .	94
9.3	Experiment . . . . .	95
<b>10</b>	<b>Conclusion</b>	<b>99</b>
	<b>Appendix A Proposed classifier: development details</b>	<b>101</b>
A.1	General notations . . . . .	101
A.2	Expression of $\mathcal{A}_{i,k}$ . . . . .	102
A.3	Expression of $Q^\square$ . . . . .	104
A.4	Expression of $F$ for equidistant targets . . . . .	105
A.5	$F$ is convex (but not strictly convex) . . . . .	106
A.6	Infinite $\gamma$ with the Inverse Function as Weight . . . . .	107



<b>Appendix B List of handcrafted features</b>	<b>109</b>
<b>Appendix C Additional figures</b>	<b>113</b>
C.1 Experiment: Influence of targets' definition on classification . . . . .	113
C.2 Parameter $\gamma_i$ with t-SNE . . . . .	115
C.3 CIFAR-10 . . . . .	116
C.3.1 $d=100$ features . . . . .	116
C.3.2 All features ( $d=1000$ ) . . . . .	116
C.4 Zooscan data set . . . . .	117
C.5 UVP5-HD data set . . . . .	120
C.6 UVP5-HD Copepods . . . . .	122
<b>Appendix D Projection of an ellipsoid: Development details</b>	<b>125</b>
D.1 Axes-align ellipsoid . . . . .	125
D.2 Deriving $S_\epsilon$ . . . . .	125
D.3 Semi-axes for perspective projection . . . . .	126
<b>Appendix E The proposed method, step-by-step</b>	<b>127</b>
E.1 Learning stage . . . . .	127
E.2 'Prediction' stage . . . . .	127
E.3 Uniformly random rotations . . . . .	128
<b>Appendix F Distribution of selected sample images</b>	<b>129</b>
<b>Bibliography</b>	<b>133</b>

## Chapter 1

### Introduction

**Key points – Plankton is crucial in the biosphere and automated processing methods for plankton imaging system are essential to address marine ecology issues**

1. Plankton organisms are very diverse.
2. They strongly contribute to Large scale biogeochemical fluxes.
3. Recent *in situ* imaging systems are efficient for the study of plankton.
4. There is a need for automatic methods to process the large amounts of data they generate.

**Chapter 1 – Introduction:**

1.1	Definitions . . . . .	2
1.2	Context . . . . .	2
1.2.1	Plankton Ecology . . . . .	2
1.2.2	Biogeochemical fluxes at the ocean surface . . . . .	4
1.3	Images of plankton . . . . .	5
1.3.1	EcoTaxa: processing millions of plankton images . . . . .	5
1.3.2	Classifying plankton images . . . . .	5
1.4	Contributions . . . . .	9

**1.1 Definitions**

Let us start with some useful definitions for the reading of this manuscript.

**Plankton:** Living organisms drifting with the current

**Phytoplankton:** Vegetal plankton

**Zooplankton:** Animal plankton

**Meso-plankton:** Plankton between  $\sim 200 \mu\text{m}$  and  $\sim 2 \text{ mm}$

**Copepod:** Small crustaceans, dominant group in the meso-plankton

**Taxonomy:** Hierarchical classification in groups

**Taxon:** Taxonomic group (*plural* taxa)

**Oceanography:** Study of the oceans

**Marine Ecology:** Ecology of the marine systems

**Biogeochemistry:** Study of the biological, physical, geological and chemical processes

*in situ* : in the environment

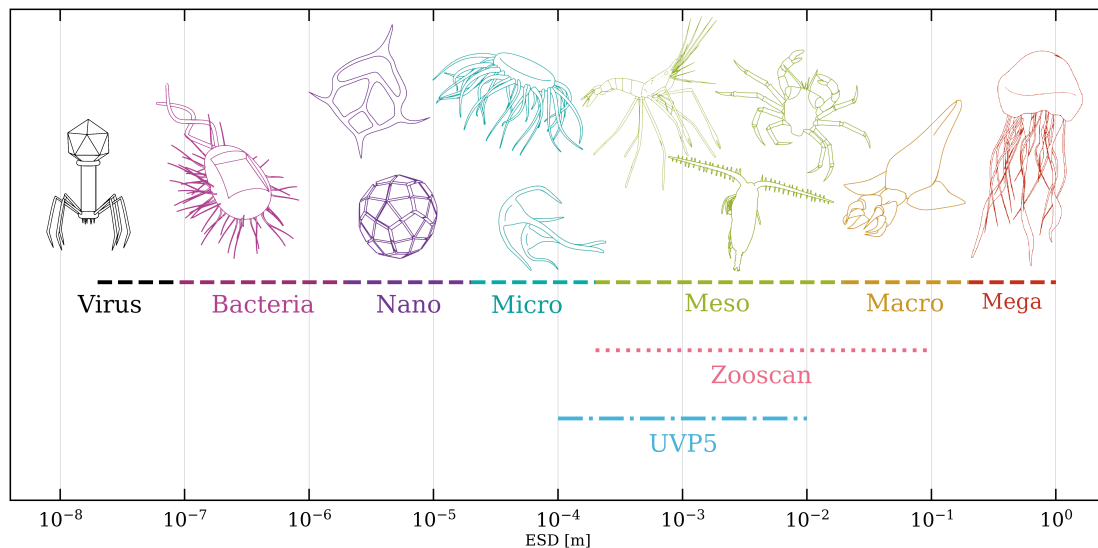
*ex situ* : out of the environment

**PgC:** Petagram Carbon ( $10^{15} \text{ gC}$ )

**1.2 Context****1.2.1 Plankton Ecology**

Plankton consists of all aquatic living organisms that drift with the currents (in both marine and fresh waters). It forms an extremely diverse community [de Vargas et al., 2015], the size spectrum it covers is very wide (ranging from  $10^{-8} \text{ m}$  to  $1 \text{ m}$ , see fig. 1.1), and its members are keystone components of Earth's biosphere. First, photosynthetic plankton is responsible for about half of the fixation of carbon dioxide from the atmosphere and therefore produces an equally large amount of dioxygen [Behrenfeld et al., 2001]. It is therefore an important contributor to the regulation of climate [Volk and Hoffert, 1985]. Second, plankton is also a critical component of many marine food webs: it directly supports some of the largest fisheries on earth, off the coast of Chile for example [Thiel et al., 2007], and some emblematic species such as corals. Finally,

because plankton simply drifts, it cannot escape the conditions of the water mass it is embedded in. This makes planktonic organisms very sensitive to environmental change. Therefore, the contribution of plankton to the important processes described above will be influenced by the changes in Earth's climate [Hays et al., 2005].

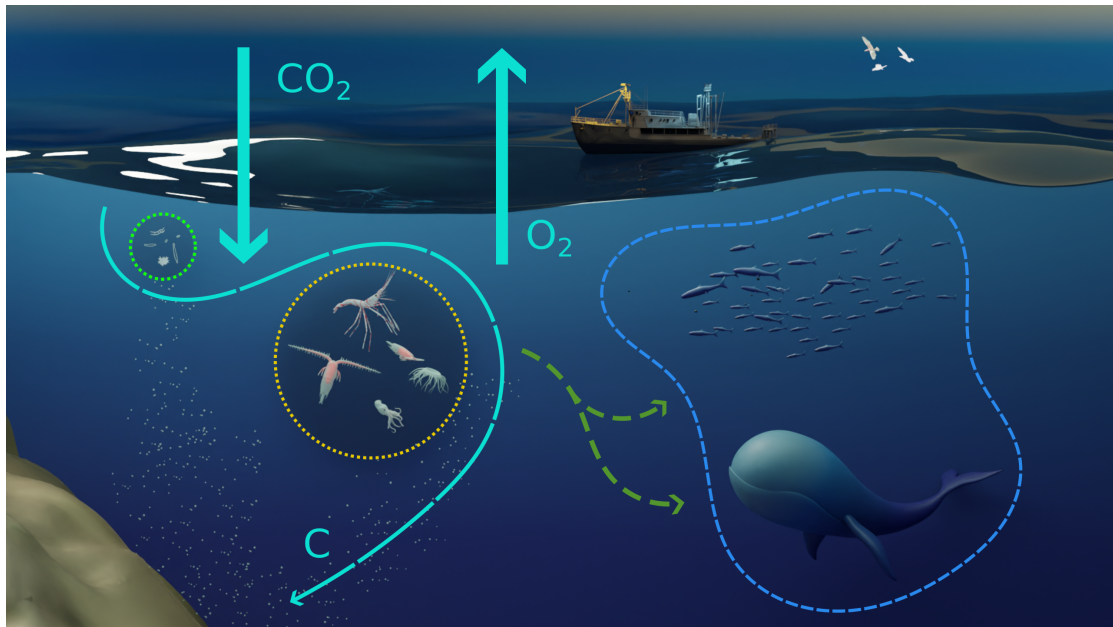


**Figure 1.1** Size range of plankton organisms, represented with their Equivalent Spherical Diameter (ESD). Figure inspired from Lombard et al. [2019]; Sunagawa et al. [2020]. In lexicographic order virus, bacteria, nano-plankton, micro-plankton, meso-plankton, macro-plankton and mega-plankton. The dotted line shows the ZooScan imaging range (see section 4.2.1), the dashed-dotted line shows the same for the UVP5 camera (see section 4.3). Those are two popular imaging instruments.

Ecology aims at studying living organisms and the interactions among themselves and with the environment. One usual first step to achieve this is the taxonomic classification of organisms. Then, relationships among these taxonomic groups and with their environment help to understand the functioning of ecosystems, including the transfer of energy from a community to another, the biomass, and the biogeochemical cycles (e.g., carbon, nitrogen). While the taxonomic classification of living beings is essential in ecology, a new approach has recently emerged that focuses on the '*functional traits*' of organisms and their interactions with their environment. Functional traits are characteristics that describe an organism's (or a community) ability to grow, survive, and reproduce, e.g., size and lipid reserves both influence all these three processes [Martini et al., 2021]. Functional approach is a complementary way to describe ecosystems and study the distribution of organisms, compare to the taxonomic one. Still, it is often challenging to measure functional traits accurately, especially in a generic way across taxa (see Orenstein et al. [2022] for a study on trait estimation from plankton images). Crossing approaches (taxonomic and functional) open the path to a new area in ecology, that is expected to produce efficient results in the near future [Martini et al., 2021]. As an overview, let us say that the ultimate goal in plankton ecology would be to have high resolution distribution of planktonic traits through space and time for each taxonomic group. Hence, the interactions of plankton organisms with their environment could be investigated.

### 1.2.2 Biogeochemical fluxes at the ocean surface

Global scale biogeochemical fluxes contribute to climate regulation. They are due to physical, geological, chemical and biological processes. In the oceans, the biological part is driven by the photosynthesis of phytoplankton that occurs at the surface. Carbon dioxide is seized from the atmosphere and dioxygen is released. Zooplankton participates in aggregating and exporting this carbon into the depths. During their life, the organisms accumulate the carbon in the phytoplankton by eating them; then, they produce fecal pellets that sink and when they die, their carcasses also sink (both contain carbon). The carbon is sequestered for hundreds to thousands of years. This cascade phenomenon generates a carbon flux from the atmosphere to the ocean floor; it defines the so-called '*biological carbon pump*' [Longhurst and Glen Harrison, 1989]. The pump is represented in the figure 1.2, together with the carbon flux in the ocean (in cyan). Multiple approaches tackle the quantification of its efficiency [Buitenhuis et al., 2013; Moriarty and O'Brien, 2013]. As an indication, an estimation [Le Quéré et al., 2015] is 2.4 billion tonnes of carbon per year (computed for the last decade). In this thesis, we focus on its relationship with the size of the zooplankton organisms.



**Figure 1.2** Schematic representation of the biological carbon pump. Figure inspired from the Kaggle competition 'National Data Science Bowl'<sup>1</sup>

A high correlation has been observed between the meso-plankton biomass (organisms between  $\sim 200 \mu\text{m}$  and  $\sim 2 \text{ mm}$  in size, see fig. 1.1 and section 1.1) and the intensity of the biological carbon flux, particularly in the surface, mixed layer [Buitenhuis et al., 2006]. The biomass of plankton is proportional to their volume through a density, that depends on the taxon. A general way of describing the relationship between plankton organisms and the biological carbon pump, is to say that plankton biomass distribution (in space) and, by extension, biovolume distribution, can be considered as a proxy to investigate carbon absorption in the ocean. Hence, producing accurate distribution of the volume of zooplankton organisms is an essential step toward the understanding of global biogeochemical fluxes. Such estimates, have been easier to achieve by

<sup>1</sup><https://www.kaggle.com/c/datasciencebowl>

the development of new *in situ* imaging instruments in recent years (see section 1.3.1 for details and [Lombard et al., 2019] for a review).

From the databases, total volume estimations can be computed as the sum of the individual volumes estimated from the 2-d *in situ* images. A common method for estimating the volume from a 2-d image is to make the use of the Equivalent Spherical Diameter (ESD) of the object. This method will be detailed in section 5.4.2, but let us already mention that it relies on the surface area of the object and, most important, that it is biased when the object is not spherical (which is often the case for plankton organisms). A comprehensive study of the total biomass of mesozooplankton through volume measurements from 2-d *in situ* images, with inference to global scale, can be found in Drago et al. [2022].

## 1.3 Images of plankton

### 1.3.1 EcoTaxa: processing millions of plankton images

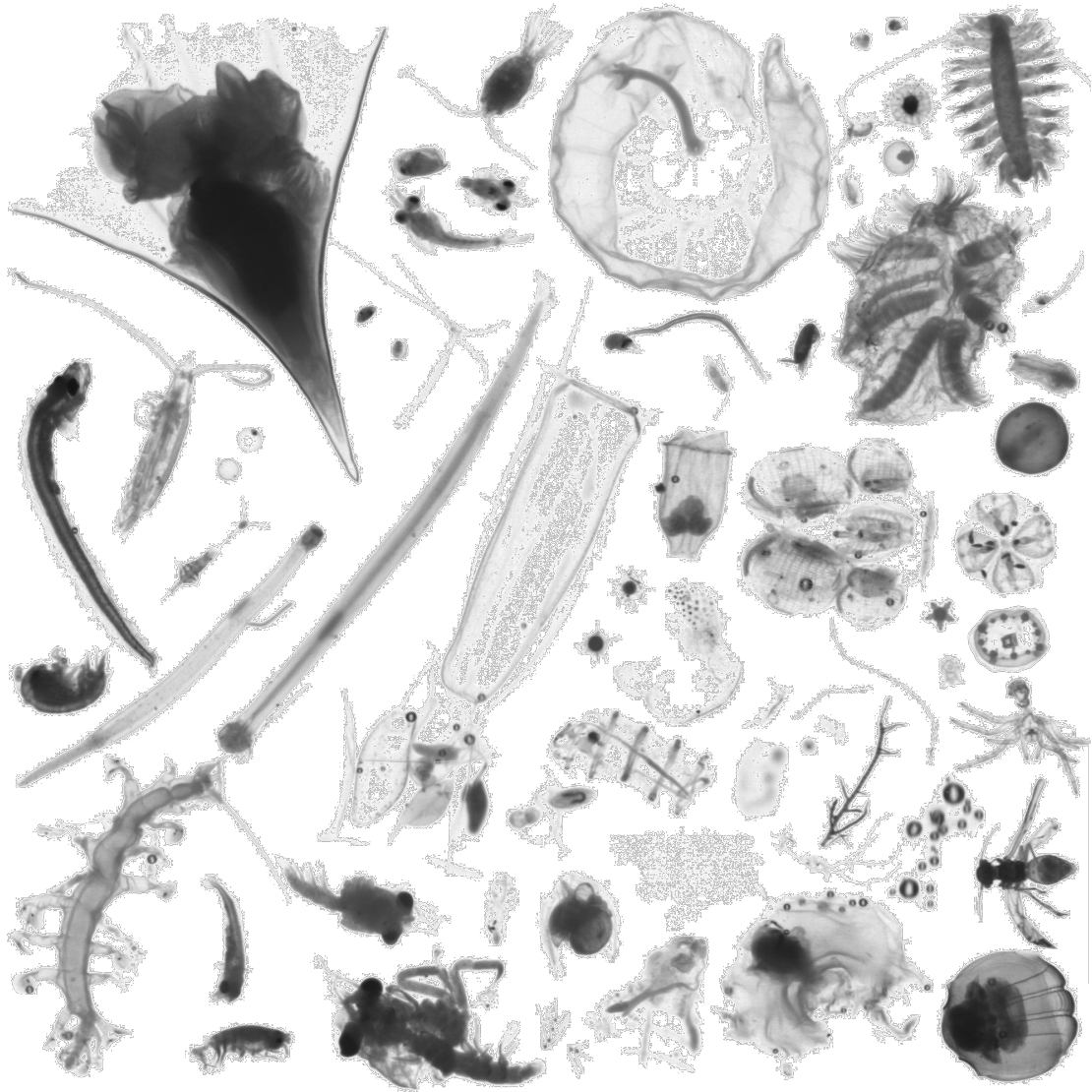
Key imagery instruments used nowadays are, the Imaging FlowCytobot (IFCB) [Sosik and Olson, 2007] (<10 to 150  $\mu\text{m}$ ), the Underwater Vision Profiler (UVP) [Picheral et al., 2010, 2022] (see section 4.3, 100  $\mu\text{m}$  to 1 cm), the ZooScan [Gorsky et al., 2010; Grosjean et al., 2004] (> 200  $\mu\text{m}$ ) and the Video Plankton Recorder [Benfield et al., 1996]. Large scale marine campaigns of *in situ* observations together with imaging systems in lab have produced (continue and will continue to produce) considerable amount of images. For example, the Underwater Vision Profiler 5 (UVP5) has been deployed in over 9,000 locations, through  $\sim 150$  cruises, organized by  $\sim 15$  countries [Kiko et al., 2022]. These data feed the EcoTaxa<sup>2</sup> [Picheral et al., 2017] web platform on a continuous basis, which has been developed specifically for this purpose. Today, it contains more than 250 million of images from around the world and thus, provides access to a wide range of data sets, essential for addressing marine environmental issues. Examples of plankton ZooScan images extracted from EcoTaxa are given in fig. 1.3.

### 1.3.2 Classifying plankton images

#### A challenging real-world case study

The classification of plankton images, manually or automatically, is challenging. There are several reasons for this; we list a few here : (i) multiple instruments are needed to cover the size spectrum of several orders of magnitude (see fig. 1.1) and each produces images with different characteristics, (ii) the morphology and the opacity of organisms are very diverse (*e.g.*, see fig. 1.3 for the zooplankton) and last but not least, (iii) the distribution of the organisms per taxonomic class is highly imbalanced. In view of the volume of data, the need for automation of data processing is obvious. In a comprehensive review on automated techniques for plankton images, Irisson et al. [2022] gives precise motivations (*e.g.*, 2 million new objects per year for the ZooScan). In recent years, several methods have been explored by the scientific community to deal with this new, complex, real application of automated image classification. In this section, we present a state of the art in plankton image classification. The purpose is to give a rough idea of the existing methods, that we distinguish mainly between those based on *handcrafted* features (section 1.3.2) and those based on Convolutional Neural Networks (CNNs) (section 1.3.2). Diverse methods are used in the literature, depending on the data at hand and the scientific goal. Let us remark that we focus our review on supervised methods (*i.e.*, classification among defined classes) since ecologists are generally interested in the taxonomic distribution of the

<sup>2</sup><https://ecotaxa.obs-vlfr.fr/>



**Figure 1.3** Images of plankton from the ZoosScan instrument (see section 4.2.1) provided by EcoTaxa. There are 34 taxonomic groups represented (2 images per group). The scale is the same for all organisms, *i.e.*, the relative size is real. The purpose of this image is to expose the diversity of planktonic organisms (in size, morphology, opacity, complexity). Note, that non-plankton objects such as algae, bubbles, dead insects are also considered, mainly represented on the bottom right of the figure.



samples, even for trait-based approaches. Moreover, there exists different taxa with similar morphologies that need to be distinguished and only supervised methods can achieve that. Nevertheless, unsupervised methods can be useful for exploring unknown data (*i.e.*, when there are no particular expectations) or detecting new classes. For this purpose, unsupervised approaches combined with manual refinement have proven to be efficient (*e.g.*, see Schröder et al. [2020]).

### Classification with *handcrafted* features

Defining a coherent distance between images is extremely challenging. Known distances (*e.g.*, Euclidean, Mahalanobis) are not meant to compare complex structures encoded into images. Instead of designing methods that rely on the (very) high dimensional image space (*e.g.*, number of pixels of the image for a 2-d image), the standard methods for classifying images consist of extracting summary features from the image as a first step, to then classify the samples in the lower dimensional feature space. In particular, for plankton, the standard is to work with one image per organism (using segmentation methods), on a homogenous background. Therefore, the features are expected to describe the object of interest. It is clear that the extraction of the features is essential and, somehow, subjective. Some basic features are: the size of the object in the image and the moments of the pixel intensity of the image (*e.g.*, mean, variance). Large improvements have been reached with Scale Invariant Feature Transform (known as SIFT) [Lowe, 1999] and, then, Bag of Words [Lazebnik et al., 2006; Saluja et al., 2022]. For plankton images, one can note the usage of co-occurrences matrices [Davis et al., 1979; Hu and Davis, 2005; Sosik and Olson, 2007].

With these image features, *machine learning* classification methods aim at classifying samples of unknown labels according to samples of known labels. Our classification framework is introduced in section 2.1. Here we focus on referencing the methods for plankton images. The first published study on plankton images that make the use of machine learning is Schlimpert et al. [1980], for microscopy imagery. Already in the late 90s, some automated classification techniques achieved human-level accuracy [Culverhouse et al., 1996; Tang et al., 1998]<sup>3</sup> Then, two sets of studies have had a particularly strong impact on the field, as evidenced by their high citation counts. The first one is based on the ZooScan instrument and the Zooprocess software [Gorsky et al., 2010; Grosjean et al., 2004], that are still in use today. The authors proposed a pipeline for imaging zooplankton, extracting features and finally classifying samples. They investigated multiples classification methods and obtained interesting performances ( $\sim 75\%$  for a few classes) with a Random Forest (RF) algorithm [Breiman, 2001]. The extracted features are mainly based on geometric descriptors and image moments; the updated list of them is given in appendix B, latter used in chapter 4. Second, the work of Sosik and Olson [2007] presented a comparable study for the IFCB instrument and investigates more feature extraction methods. This time, a Support Vector Machine (SVM) [Cortes and Vapnik, 1995] was used to achieve high accuracy. The work of Benfield et al. [2007] gives an overview of the State-Of-The-Art in plankton image classification at that time. The conclusion is that good accuracies (*i.e.*, 70-80%) are achieved for samples distributed into 10 to 20 classes. Since then, most of the works based on handcrafted features attempts to apply such classification methods in ecological studies.

A main conclusion (that also holds for the next section section 1.3.2) of a recent review [Irisson et al., 2022] is that understanding the actual State-Of-The-Art in real-world plankton image automatic classification is difficult. This is mainly due to (*i*) the lack of reference data

---

<sup>3</sup>The former use a multilayer perceptron, but is referenced here since it does not make use of convolutional layers. Still, it is interesting to note that one of the first impressive results was produced by Artificial Neural Networks (ANNs).

sets to compare different methods consistently and (ii) the evolution of the complexity of the classification task (mainly the number of classes) with the progression of the classifiers over time.

### Convolutional Neural Networks: breakthroughs & limitations

As for many image applications domains, the introduction of CNNs revolutionized the field [Rawat and Wang, 2017; Krizhevsky et al., 2017, 2012]. For plankton images, it crystallized around the National Data Science Bowl Kaggle competition in 2015<sup>4</sup>. Since then, multiples works were based on CNNs. The major contribution of the CNN framework is that it offers the possibility to optimize the features' extraction together with the classification. While it is largely accepted, this last statement is not completely accurate. As we will see in chapter 2, the CNN optimization for classification is actually a regression. Nevertheless, it is an alternative to the manual and subjective definition of handcrafted features, and has proven to be very useful for multiple image classification applications. For plankton images, it brings classification performance further, by discriminating finer taxonomic groups (up to more than 100 classes, *e.g.*, Luo et al. [2018]). An interesting, unexpected point is that patterns caught by CNNs methods seem to be similar among various type of image data sets. Indeed, CNNs that have been optimized on large and diverse enough data sets can produce relevant features for other applications than the one they were trained for. This is known as '*transfer learning*' [Weiss et al., 2016]. Note, it is also often used for initiating the optimization of a specific classification task, which is then referenced as '*fine tuning*'. A comprehensive study that shows the interest of transfer learning for plankton image applications is Orenstein and Beijbom [2017].

As mentioned above, the improvements made by ANNs methods are huge. However, they also have drawbacks. Indeed, our understanding of such networks remains poor. It has often been said that ANNs are '*black-boxes*' and, even with the developments of remarkable work on the interpretation of such models by the scientific community (see Kutyniok [2022]), the previous affirmation remains true. More precisely, ANNs excels at differentiating high-dimensional samples (*e.g.*, images) from different classes, but we are still unable to explain *why*. Misunderstanding their functioning can lead to unexpected, problematic situations. A well-known case is that of *adversarial attacks*, which cause drastic changes in performance by adding a simple and imperceptible (at least to humans) change to the input. The absence of a comprehensible framework is confusing. Ecologists, among others, need reliable, efficient, and trustable models to produce replicable studies. Today, in the case of real-world applications, the confidence in the results of ANN is based on the subjective and incomplete *tests* carried out by the user, which, by construction, do to cover all cases. The existence of strong theoretical results on ANNs is missing. It would add a new dimension to the confidence placed in these impressive methods and provide insights for new outcomes. As an illustration, two improvements that would be significant are (i) the understanding of the generalization error<sup>5</sup> (*e.g.*, determination of a bound) and (ii) the proof of convergence to a global minimum of the network optimization<sup>6</sup>. To conclude, there is a lack of theoretical foundations for ANNs, compared to 'standard' approaches [Vapnik, 1999] and, in practice, this can mean a lack of confidence or a misunderstanding of the results. To go a step further, the *preprint* Kutyniok [2022] addresses those questioning in detail.

<sup>4</sup><https://kaggle.com/c/datasciencebowl>

<sup>5</sup>*i.e.*, the expected error on unknown data, see details in section 2.1.2

<sup>6</sup>see Chizat and Bach [2018] for the convergence of a one-hidden layer ANN

## 1.4 Contributions

The development of modern machine learning and image processing methods is essential to meet the challenges of marine ecology. In this context, the aim of this manuscript is to present the research work done during this thesis, which address both machine learning and plankton ecology questioning, with 2-d images as a common denominator. The contributions are distinguished into two main parts. In the first one, we establish a result related to the standard Nearest-Neighbour (NN) classification and apply the resulting method to the classification of 2-d plankton images. More precisely, in chapter 2, we define a sample transformation based on class *targets*, for the classification. We study the influence of the targets positions and demonstrate the equivalence to the Weigthed-Nearest-Neighbours (WNN) classifier for a specific choice of targets. In this case, we define a kernel associated to WNN with the transformation. We propose a modern implementation (with fast Leave-One-Out Cross-Validation ( $CV_{LOO}$ ) predictions) of the resulting WNN classifier in chapter 3. We conclude this part with chapter 4, by showing that the implementation can handle large data sets and help to produce reliable results for real-world application such as 2-d plankton images. In a second part, focusing on the most abundant taxonomic group, namely, the copepods; we tackle the estimations of the total biovolume of copepods from 2-d *in situ* images in chapter 5. First, in chapter 6 we propose an image processing procedure to extract the body of the copepods, which more appropriate to estimate the volume. Second, in chapter 7) we highlight the biases of the standard methods due to the projection of the organisms onto the image plane. With the help of a geometrical modelling, we define a simulation pipeline for correcting the total copepods' volume estimates. Finally, in chapter 8 we apply the correction to a real data set of 2-d *in situ* copepods images. In chapter 9 we give an additional application with the simulator. We end this manuscript with a general conclusion.



## **Part I**

# **A Nearest-Neighbours Kernel for Classification: a case study of Plankton Images**



## Chapter 2

### A Kernel Associated to the Nearest-Neighbours

#### Key points – Proposition of a classifier based on a supervised transformation

1. ANNs transform samples to a *target* space. The classification is done by the *nearest-target* classifier.
2. We propose a supervised classification method that relies on a non-linear transformation guided by the *targets* and the *nearest-target* classifier.

#### Contributions – Characterization of the proposed classifier

3. Definition of the nearest-target classifier and the proposed transformation to target space.
4. Study of the influence of targets positions for two and more classes
5. Conjecture on *optimal* targets for more than two classes
6. Demonstration of the equivalence to a *weighted-k-nearest-neighbour* classifier for equidistant targets.
7. Definition of a *kernel* associated to the *weighted-k-nearest-neighbour* classifier.



**Chapter 2 – A Kernel Associated to the Nearest-Neighbours:**


---

2.1	Reminders on supervised classification . . . . .	14
2.1.1	Empirical Risk Minimization . . . . .	14
2.1.2	Evaluating the generalization performances . . . . .	15
2.2	Motivations . . . . .	16
2.3	Proposed Classifier . . . . .	17
2.3.1	Definition . . . . .	17
2.3.2	Additional Notations . . . . .	18
2.3.3	Properties . . . . .	19
2.4	Influence of Target Positions . . . . .	20
2.4.1	Problem Statement . . . . .	20
2.4.2	Case 1/3: $\hat{k} \neq m$ and $l \neq m$ (applies only for $p \geq 3$ ) . . . . .	21
2.4.3	Case 2/3: $\hat{k} = m$ (consequently, $l \neq m$ ) . . . . .	22
2.4.4	Case 3/3: $l = m$ (consequently, $\hat{k} \neq m$ ) . . . . .	23
2.5	Optimal Targets for 3 Classes or More . . . . .	24
2.5.1	Variational Formulation . . . . .	24
2.5.2	Minimization of $F$ . . . . .	25
2.5.3	Classification Point-of-View . . . . .	27
2.6	Selection of $\gamma$ . . . . .	28
2.6.1	Limit Cases . . . . .	28
2.6.2	About an Optimal Value . . . . .	30
2.7	Some Choices of parameters $\gamma_i, i \in [1..n]$ . . . . .	31

---

**2.1 Reminders on supervised classification****2.1.1 Empirical Risk Minimization**

Supervised classification aims at finding a classification function (or simply a *classifier*) that associates a label  $y \in \mathcal{Y}$  among  $p$  pre-defined labels ( $\mathcal{Y} = \{1 \dots p\}$ ) to a sample  $x \in \mathcal{X} \subset \mathbb{R}^d$ , based on a set of pairs of samples and their associated labels, *i.e.*, a data set  $\mathcal{S} = \{(x_1, y_1) \dots (x_n, y_n)\}$  (see an example in fig. 2.1). The observations  $\{x_i, y_i\}_{i=1}^n$  are drawn from the underlying, unknown, distribution  $\mathcal{P}(x, y)$ . More precisely, we consider a class of functions  $f \in \mathcal{F}$ ,  $\mathcal{F}$  the space of functions considered. For the supervised classification, the reference point is the Bayes classifier, defined as

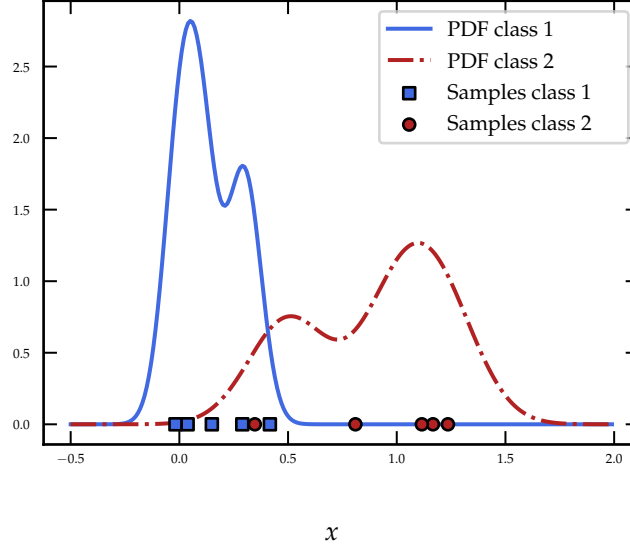
$$f^{\text{Bayes}}(x) = \arg \max_{k \in \{1 \dots p\}} \mathcal{P}(y = k|x). \quad (2.1)$$

This optimal classifier can not be used in practice since it relies on the unknown class distributions  $\mathcal{P}(x|y = k)$ . Instead, other classification functions have been proposed. In order to evaluate them, we need to introduce a measure of quality. Let us consider the misclassification loss function

$$l(f(\cdot), y) = \begin{cases} 0, & \text{if } f(\cdot) = y \\ 1, & \text{otherwise} \end{cases} \quad (2.2)$$

and the associated *risk*

$$\mathcal{R}(f) = \int_{\mathcal{X} \times \mathcal{Y}} l(f(x), y) d\mathcal{P}(x, y). \quad (2.3)$$



**Figure 2.1** Illustration of the distributions  $\mathcal{P}(x|y = 1)$  and  $\mathcal{P}(x|y = 2)$  for a binary classification setting in one dimension. The blue squares and red circles are examples of samples.

Then, the *learning* process consists of solving

$$\inf_{f \in \mathcal{F}} \mathcal{R}(f), \quad (2.4)$$

that is, searching for the function  $f$  that minimize the risk.

However, in practice, the evaluation of the risk is not possible (recall  $\mathcal{P}(x, y)$  is unknown). Instead, let us we define the (computable) *empirical risk*

$$\mathcal{R}_{\text{emp}}(f) = \frac{1}{n} \sum_{i=1}^n l(f(x_i), y_i). \quad (2.5)$$

This time, the goal is to find the optimal function  $f$  that minimize the empirical risk, which defines the Empirical Risk Minimization (ERM) principle

$$\min_{f \in \mathcal{F}} \mathcal{R}_{\text{emp}}(f). \quad (2.6)$$

that gives the relative number of misclassified samples among all the data set.

In practice, the *accuracy* (ACC) is the metric generally used to judge the classification performances and is directly linked to the empirical classification risk by

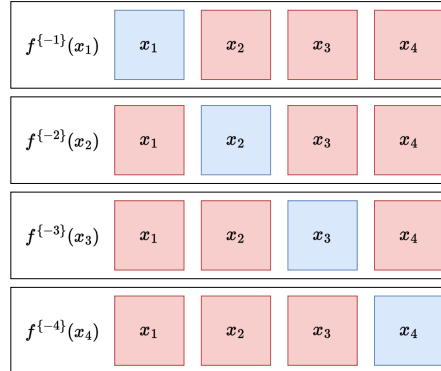
$$\text{ACC} = 1 - \mathcal{R}_{\text{emp}}^{\text{clf}} \quad (2.7)$$

### 2.1.2 Evaluating the generalization performances

Assessing whether a classifier will be accurate on new data cannot rely on the minimization of the empirical. Indeed, the overall goal is to be able to classify samples unseen during training, meaning to minimize the error of *generalization*, that is not accessible. Instead, a common

approach is to derive statistical bounds on the deviation of the empirical risk (eq. (2.5)) from the risk (eq. (2.3)), as a function of the number of samples  $n$ . It is clear that lower deviations are preferable to expect better generalization performances. Such bounds are referenced for standards classification methods such as SVM [Vapnik, 1999] or Weighed-Nearest-Neighbours (WNN) [Samworth, 2012].

In practice, with the data at hand, one uses Cross-Validation (CV) techniques [Stone, 1974], to evaluate the generalization performances of a given method. One approach (probably the most used) consists of dividing the data set into  $k$  subsets, learning from  $k - 1$  subsets and testing on the remaining subset to estimate a generalization error, then averaging the  $k$  errors obtained when dealing with each of the possible learning/testing splits. This procedure is called the  $k$ -fold CV. Classifiers usually depend on parameters that are optimized with a grid search for parameter combinations (say  $m$  feasible combinations). The overall optimal parameters are the ones that minimize the generalization error. This last can be done with CV. Then, for each fold, the empirical risk is estimated  $m$  times. Except for classifiers that do not require optimization during learning (e.g.,  $k$ -Nearest-Neighbours ( $k$ -NN) [Cover and Hart, 1967]), the overall evaluation can be time-consuming, specially for large values of  $k$  and/or  $m$ . On the other hand, considering many folds reflects better the underlying generalization performances, since the learning rounds rely on more data [Elisseff et al., 2003]. In particular, the limit case  $k = n$  is called the Leave-One-Out Cross-Validation ( $CV_{LOO}$ ). An illustration is given in fig. 2.2 for  $n = 4$ .



**Figure 2.2** Scheme view of the  $CV_{LOO}$  principle for  $n = 4$ . The red squares represent the training samples for the classifier  $f^{(-i)}$  and the blue square represent the test sample. The accuracy is computed for each of the four predictions, and the final score is taken as the mean.

## 2.2 Motivations

The model proposed in this chapter rely on one key observation about ANNs. Here, we present the functioning of such classification methods. The simpler ANN is the Perceptron [Mcculloch and Pitts, 1943], a linear, binary classifier. For classes  $y \in \{0, 1\}$ , it is defined as

$$f^P(x) = H(w \cdot x + b) = \begin{cases} 1 & \text{if } \sum_i^d w_i x_i + b > 0 \\ 0 & \text{otherwise} \end{cases}, \quad (2.8)$$

where  $H$  is the Heaviside function, the  $w_i$ s are the *weights* and  $b$  is the bias, that are the parameters of the model (to optimize). The resulting model is an optimal parametric transformation, which perform dimension reduction. The key observation is: *the classification decision of a perceptron is a*

simple threshold on a transformation of the input sample ( $w \cdot x + b$  here). This means there are two stages in estimating the label of a sample. First, it is transformed from the input space to a *target space*. Two, the classification is given by a simple threshold, in this target space.

The extension to the multi-class ANNs is done by defining  $p$  targets, one per class. Then, the classification is defined as the Nearest Target (see section 2.3.1 for a definition), which predict the sample label as the one of the nearest target in the target space (that includes the previous binary case). In practice, for the ANNs, the targets are typically defined as the canonical basis of  $\mathbb{R}^p$ . This can be motivated by the use of loss functions coming from the probability theory, such as the *cross-entropy*. Nevertheless, to the best of our knowledge, there is no other objective motivation. The ANN approach to classification (among which the popular CNNs) can be seen as a transform-then-classify-simply procedure.

While the behaviour of ANNs is not easy to understand (see section 1.3.2), the idea of transforming samples around targets to then classify them easily is interesting to investigate<sup>1</sup>.

## 2.3 Proposed Classifier

### 2.3.1 Definition

The proposed classifier relies on some vectors, called targets, associated with each class. To classify a sample, it is transformed into a target-compatible vector. This transformed sample is then compared with the class targets in order to take the classification decision.

Let us recall that the learning samples are distinct vectors  $x_i, i \in [1..n]$ , of  $\mathbb{R}^d$ , distributed among  $p$  classes.

**Definition 2.1. Proposed Classifier.** We define targets as distinct vectors  $T_k, k \in [1..p]$ , of  $\mathbb{R}^e$  where  $T_k$  represents class  $k$ . The target associated with the learning sample  $x_i$  is  $t_i = T_k$  if  $x_i$  belongs to class  $k$ . For any  $x \in \mathbb{R}^d$ , the proposed classifier is defined as

$$\hat{k}(x) = \arg \min_{k \in [1..p]} |u(x) - T_k|^2 \quad (2.9)$$

where the so-called transformed sample  $u(x)$  is defined as

$$u(x) = \sum_{i=1}^n w_i^s(x) t_i \quad (2.10)$$

and  $w_i^s(x), i \in [1..n]$ , are some positive weights such that

$$\sum_{i=1}^n w_i^s(x) = 1. \quad (2.11)$$

These weights are detailed in definition 2.2.

**Definition 2.2. Proposed Weights.** For two samples  $x$  and  $y$  of  $\mathbb{R}^d$ , the weighting function is defined as

$$w_\gamma(x, y) = w_\gamma^{\text{rad}}(|x - y|) = \gamma w^{\text{rad}}(\gamma|x - y|), \quad (2.12)$$

where  $\gamma$  is a positive constant, and  $w^{\text{rad}}$  is, on  $\mathbb{R}^+$ , continuous, finite, positive (thus it has an infinite support), monotonically decreasing, and has a limit of zero at infinity.

<sup>1</sup>Note the similitude with the *kernel trick*, largely used with SVMs (but without targets).

Then, for  $j \in [1..n]$ , we choose

$$w_j^s(x) = \alpha(x)w_{\gamma_j}(x, x_j), \quad (2.13)$$

where  $\alpha(x)$  is such that eq. (2.11) is true and  $\gamma_j$  is a positive constant tuning locally the weighting function 'width'. Note that eq. (2.11) has a sense only if the weights are not all equal to zero. This is guaranteed since  $w^{\text{rad}}$  has an infinite support. So finally,

$$w_j^s(x) = \frac{w^{\text{rad}}(\gamma_j|x - x_j|)}{\sum_{i=1}^n w^{\text{rad}}(\gamma_i|x - x_i|)}. \quad (2.14)$$

### 2.3.2 Additional Notations

**Sample-to-class function:**  $c$

- Learning sample class index:  $c_i = c(x_i)$ .
- Class indicator matrix:

$$C = [e_{c_1}^p e_{c_2}^p \cdots e_{c_n}^p], \quad (2.15)$$

where  $e_i^p$  is the  $i$ th element of the canonical basis in  $\mathbb{R}^p$ .

**Targets**

- Target set:  $\{T_1, \dots, T_p\}$ .
- Target vector:

$$U = [U_1^\top U_2^\top \cdots U_e^\top]^\top \quad (2.16)$$

$$\text{where } [U_1 U_2 \cdots U_e] = [T_1 T_2 \cdots T_p]^\top. \quad (2.17)$$

**Weights**

- Per-learning-sample weight:  $w_j^s(i) = w_j^s(x_i)$ .
- Per-sample weight matrix (not symmetric in general):

$$\Omega^s = [w_j^s(i), (i, j) \in [1..n]^2]. \quad (2.18)$$

- Per-class weight: for a sample  $x$  and  $k \in [1..p]$ ,

$$w_k^c(x) = \sum_{t_i=T_k} w_i^s(x). \quad (2.19)$$

- Per-class weight of learning samples:  $w_k^c(i) = w_k^c(x_i)$ .
- Per-class weight matrix:

$$\Omega^c = [w_k^c(i), (i, k) \in [1..n] \times [1..p]] = C\Omega^s. \quad (2.20)$$

**Transformed learning sample:**  $u_i = u(x_i)$ .

As a convention, indices  $i$  and  $j$  are used for learning samples and indices  $k, l$ , and  $m$  are used for classes. Other, general notations used here are defined in appendix A.1.

### 2.3.3 Properties

**Property.** For finite  $\gamma_j, j \in [1..n]$ , and for all  $x \in \mathbb{R}^d$ , the weight  $w_j^s(x)$  belongs to the interval  $]0, 1[$ . At infinity, we focus on the case  $\gamma_j = \gamma \forall j$ , and we have

$$\begin{cases} \lim_{\gamma \rightarrow +\infty} w_j^s(j) = 1 \\ \lim_{\gamma \rightarrow +\infty} w_j^s(i) = 0 \forall i \in [1..n], i \neq j \end{cases} \quad (2.21)$$

*Proof.* For finite  $\gamma_j$ , the property follows from the positivity of  $w^{\text{rad}}$  and the constraint (2.11).

At infinity, we have

$$(2.14) \Leftrightarrow w_j^s(x) = \frac{1}{1 + \sum_{\substack{i=1 \\ i \neq j}}^n \frac{w^{\text{rad}}(\gamma|x-x_i|)}{w^{\text{rad}}(\gamma|x-x_j|)}}. \quad (2.22)$$

Then, it can be checked that eq. (2.21) is true.  $\square$

**Property.** For finite  $\gamma_j, j \in [1..n]$ , the situation

$$\forall (k, i) \in [1..p] \times [1..n], \begin{cases} w_k^c(i) = 1 \text{ if } t_i = T_k \\ w_k^c(i) = 0 \text{ if } t_i \neq T_k \end{cases} \quad (2.23)$$

cannot happen.

*Proof.* The situation (2.23) means that, for a learning sample  $x_i$  which belongs to the class  $k$ , 'all the weight' is put on the samples of that class, disregarding the samples of the other classes. Since the weighting function  $w^{\text{rad}}$  has an infinite support, it cannot happen.

For the sake of curiosity, let us check the case of weighting functions with a bounded support and all  $\gamma_j, j \in [1..n]$ , equal to a unique value  $\gamma$ . The situation (2.23) can only happen if the learning samples are well separated by class. More precisely, if  $w_\gamma$  has a bounded support with radius  $\rho$ , then we must have that any two learning samples from different classes be at least at a distance  $\rho$  from each other.  $\square$

**Property.** It can be checked that  $\sum_{k=1}^p w_k^c(x) = 1$  for all  $x$ .

**Property.** The transformed sample (2.10) can be rewritten as follows

$$u(x) = \sum_{k=1}^p w_k^c(x) T_k. \quad (2.24)$$

*Proof.* From eq. (2.10), we have

$$u(x) = \sum_{k=1}^p \sum_{\substack{i \\ t_i = T_k}} w_i^s(x) t_i = \sum_{k=1}^p \sum_{\substack{i \\ t_i = T_k}} w_i^s(x) T_k. \quad (2.25)$$

$\square$

**Claim 1.** If the targets are chosen to be the canonical basis  $e_i^p, i \in [1..p]$ , of  $\mathbb{R}^p$ , then the proposed classifier (see definition 2.1) is a weighted nearest-neighbour (WNN) classifier.

*Proof.* For all  $x \in \mathbb{R}^d$  and all  $k \in [1..p]$ , if  $T_k = e_k^p$ , we have

$$|u(x) - T_k|^2 = \left| \sum_{i=1}^n w_i^s(x) t_i - e_k^p \right|^2 \quad (2.26)$$

$$= \left| \sum_{l=1}^p w_l^c(x) e_l^p - e_k^p \right|^2 \quad (2.27)$$

$$= \left| \text{Vec}_{l=1}^p(w_l^c(x)) - e_k^p \right|^2 \quad (2.28)$$

$$= \sum_{\substack{l \\ l \neq k}} (w_l^c(x))^2 + (w_k^c(x) - 1)^2 \quad (2.29)$$

$$= \sum_{l=1}^p (w_l^c(x))^2 - (w_k^c(x))^2 + (w_k^c(x) - 1)^2 \quad (2.30)$$

$$= \sum_{l=1}^p (w_l^c(x))^2 + 1 - 2w_k^c(x). \quad (2.31)$$

Then,

$$\hat{k}(x) = \arg \min_{k \in [1..p]} |u(x) - T_k|^2 \quad (2.32)$$

$$= \arg \max_{k \in [1..p]} w_k^c(x), \quad (2.33)$$

which effectively amounts to selecting the class whose learning samples accumulate the highest  $x$ -related weight.  $\square$

**Claim 2. Nearest-Neighbour Kernel** *The kernel defined by the dot product of two transformed samples  $u(x)$  and  $u(x')$  with the canonical basis  $e_i^p$ ,  $i \in [1..p]$ , of  $\mathbb{R}^p$  as targets is called the kernel associated to the Nearest-Neighbours*

$$K_{\text{NN}}(x, x') = \langle u(x), u(x') \rangle, \quad (2.34)$$

where  $u(x) = \sum_{l=1}^p w_l^c(x) e_l^p$  is the transformation associated to the weighted nearest-neighbour classifier (see Claim 1). For brevity, we refer to it as a 'Nearest-Neighbours kernel'.

In the following, we will give elements allowing to conjecture that the canonical basis of  $\mathbb{R}^p$  is an optimal target choice in some sense. An actual proof is left for future research.

## 2.4 Influence of Target Positions

### 2.4.1 Problem Statement

To study the influence of the target positions on the performances of the proposed classifier, it suffices to fix all the targets, then randomly select one target, say  $T_m$ , move it, and study the influence of this change on classification. If there is an influence, then it can be concluded that the classification performances depend on the target positions. If not, then the two configurations, before and after moving  $T_m$ , are equivalent. Any two target configurations can then be linked together through such elementary 'jumps' between equivalent configurations, which makes them equivalent. It can therefore be concluded that the classification performances do not depend on the target positions.

The class  $\hat{k}(x)$  ( $\hat{k}$  for simplicity here) assigned to a sample  $x$  by the proposed classifier is such that

$$\forall l \neq \hat{k}, |u(x) - T_{\hat{k}}|^2 < |u(x) - T_l|^2. \quad (2.35)$$

The displacement of  $T_m$  can be modeled as

$$T'_m = MT_m, M \in \mathbb{R}^{e \times e}. \quad (2.36)$$

For convenience, we define  $T'_l$  for all  $l$  where  $T'_l = T_l$  for  $l \neq m$ . Let  $u'$  denote the sample transform defined using  $T'_m$  in place of  $T_m$ . Then,

$$u'(x) = \sum_{\substack{i \\ t_i \neq T_m}} w_i^s(x) t_i + \sum_{\substack{i \\ t_i = T_m}} w_i^s(x) MT_m \quad (2.37)$$

$$\begin{aligned} &+ \sum_{\substack{i \\ t_i = T_m}} w_i^s(x) T_m - \sum_{\substack{i \\ t_i = T_m}} w_i^s(x) T_m \\ &= u(x) + w_m^c(x)(M - I_e)T_m \end{aligned} \quad (2.38)$$

$$= u(x) + U_m(x). \quad (2.39)$$

The classification of  $x$  is not influenced by the displacement of  $T_m$  if the condition (2.35) still holds after displacement, that is

$$\forall l \neq \hat{k}, |u'(x) - T'_{\hat{k}}|^2 < |u'(x) - T'_l|^2. \quad (2.40)$$

**Claim 3.** *The positions of the targets have no influence on the classification performances in the two-class case. On the contrary, they may have an influence when there are three classes or more.*

*Proof.* For two classes, sections 2.4.3 and 2.4.4 show that inequality (2.40) holds for any target displacement matrix  $M$ . For three classes or more, sections 2.4.2 to 2.4.4 show that some choices of  $M$  can break inequality (2.40).  $\square$

#### 2.4.2 Case 1/3: $\hat{k} \neq m$ and $l \neq m$ (applies only for $p \geq 3$ )

**Claim 4.** *Inequality (2.40) can be broken by writing the target displacement matrix  $M$  as  $\alpha N$ , where  $\alpha$  is a positive constant, and letting  $\alpha$  grow arbitrarily large.*

*Proof.* In this case,  $T'_{\hat{k}} = T_{\hat{k}}$  and  $T'_l = T_l$ . Then,

$$(2.40) \Leftrightarrow |u'(x) - T_{\hat{k}}|^2 < |u'(x) - T_l|^2 \quad (2.41)$$

$$\Leftrightarrow |u(x) + U_m(x) - T_{\hat{k}}|^2 < |u'(x) - T_l|^2 \quad (2.42)$$

$$\begin{aligned} \Leftrightarrow |u(x) - T_{\hat{k}}|^2 + |U_m(x)|^2 + 2\langle U_m(x), u(x) - T_{\hat{k}} \rangle \\ < |u'(x) - T_l|^2 \end{aligned} \quad (2.43)$$

$$\begin{aligned} \Leftrightarrow |u(x) - T_{\hat{k}}|^2 + 2\langle U_m(x), u(x) - T_{\hat{k}} \rangle \\ < |u(x) - T_l|^2 + 2\langle U_m(x), u(x) - T_l \rangle \end{aligned} \quad (2.44)$$

$$\begin{aligned} \Leftrightarrow |u(x) - T_l|^2 > |u(x) - T_{\hat{k}}|^2 \\ + 2w_m^c(x)\langle (M - I_e)T_m, T_l - T_{\hat{k}} \rangle. \end{aligned} \quad (2.45)$$

---

<sup>2</sup>Let us ignore equidistance.



The sign of the inner product in eq. (2.45) depends on the relative positions of  $T_l$ ,  $T_{\hat{k}}$ , and  $(M - I_e)T_m$ . If it is negative or equal to zero, then the inequality always holds. Otherwise, let us remind that  $w_m^c(x)$  is never equal to zero for a finite value of  $\gamma_i$ ,  $i \in [1..n]$ . Let us write  $M$  as  $\alpha N$ , where  $\alpha$  is a positive constant. Then, the right-hand side term of eq. (2.45) can be made arbitrarily large by increasing  $\alpha$ , which will eventually break the inequality.  $\square$

### 2.4.3 Case 2/3: $\hat{k} = m$ (consequently, $l \neq m$ )

**Claim 5.** *With three classes or more, inequality (2.40) can be broken by choosing the target displacement matrix  $M$  such that*

$$MT_m = T_m + \frac{T_l - u(x)}{w_m^c(x)}. \quad (2.46)$$

*Proof.* In this case,  $T'_k = MT_m$  and  $T'_l = T_l$ . Then,

$$(2.40) \Leftrightarrow |u(x) - T_l|^2 > |u(x) - MT_m|^2 + 2w_m^c(x) \langle (M - I_e)T_m, T_l - MT_m \rangle. \quad (2.47)$$

We also have

$$|u(x) - MT_m|^2 = |u(x) - T_l - (MT_m - T_l)|^2 \quad (2.48)$$

$$= |u(x) - T_l|^2 + |MT_m - T_l|^2 - 2 \langle u(x) - T_l, MT_m - T_l \rangle \quad (2.49)$$

$$= |u(x) - T_l|^2 + \langle MT_m - T_l, MT_m - T_l - 2(u(x) - T_l) \rangle \quad (2.50)$$

$$= |u(x) - T_l|^2 + \langle MT_m - T_l, MT_m + T_l - 2u(x) \rangle \quad (2.51)$$

So

$$(2.47) \Leftrightarrow \langle MT_m - T_l, MT_m + T_l - 2u(x) \rangle - 2w_m^c(x) \langle (M - I_e)T_m, MT_m - T_l \rangle < 0 \quad (2.52)$$

$$\Leftrightarrow \langle MT_m - T_l, [(1 - 2w_m^c(x))M + 2w_m^c(x)I_e]T_m + T_l - 2u(x) \rangle < 0. \quad (2.53)$$

If we choose the target displacement  $M$  such that

$$MT_m = T_m + \frac{T_l - u(x)}{w_m^c(x)}, \quad (2.54)$$

then one can check that the inner product in eq. (2.53) is equal to  $|MT_m - T_l|^2$ , which breaks the inequality.

Note that this target displacement cannot be used when there are only two classes. Indeed (let us set  $l = 1$  and  $m = 2$ ),

$$MT_2 = T_2 + \frac{T_1 - u(x)}{w_2^c(x)} \quad (2.55)$$

$$= T_2 + \frac{T_1 - w_1^c(x)T_1 - w_2^c(x)T_2}{w_2^c(x)} \quad (2.56)$$

$$= T_2 + \frac{(1 - w_1^c(x))T_1 - w_2^c(x)T_2}{w_2^c(x)} = T_1, \quad (2.57)$$

thus moving the target onto the other one (which of course is *illegal*). See Claim 6 for the two-class case.  $\square$

**Claim 6.** *With two classes, inequality (2.40) holds for any target displacement matrix  $M$  for samples belonging to the class of the displaced target ( $\hat{k} = m$ ).*

*Proof.* Let us set  $l = 1$  and  $m = 2$ . Then,

$$(2.53) \Leftrightarrow \langle MT_2 - T_1, [(1 - 2w_2^c(x))M + 2w_2^c(x)I_e]T_2 + T_1 - 2w_1^c(x)T_1 - 2w_2^c(x)T_2 \rangle < 0 \quad (2.58)$$

$$\Leftrightarrow \langle MT_2 - T_1, (1 - 2w_2^c(x))MT_2 + (1 - 2w_1^c(x))T_1 \rangle < 0 \quad (2.59)$$

$$\Leftrightarrow (1 - 2w_2^c(x))|MT_2 - T_1|^2 < 0. \quad (2.60)$$

Since the sample  $x$  belongs to class 2, then  $w_2^c(x) > 1/2$ , and therefore  $1 - 2w_2^c(x) < 0$ . So eq. (2.60) indeed holds for any  $M$ .  $\square$

#### 2.4.4 Case 3/3: $l = m$ (consequently, $\hat{k} \neq m$ )

**Claim 7.** *With three classes or more, inequality (2.40) can be broken by choosing the target displacement matrix  $M$  such that*

$$MT_m = -T_m - \frac{T_{\hat{k}} - u(x)}{w_m^c(x)}. \quad (2.61)$$

*Proof.* In this case,  $T_{\hat{k}}' = T_{\hat{k}}$  and  $T_l' = MT_m$ . Then,

$$(2.40) \Leftrightarrow |u(x) - MT_m|^2 > |u(x) - T_{\hat{k}}|^2 + 2w_m^c(x)\langle (M - I_e)T_m, MT_m - T_{\hat{k}} \rangle \quad (2.62)$$

$$\Leftrightarrow \langle MT_m - T_{\hat{k}}, MT_m + T_{\hat{k}} - 2u(x) \rangle > 2w_m^c(x)\langle (M - I_e)T_m, MT_m - T_{\hat{k}} \rangle \quad (2.63)$$

$$\Leftrightarrow \langle MT_m - T_{\hat{k}}, MT_m + T_{\hat{k}} - 2u(x) - 2w_m^c(x)(M - I_e)T_m \rangle > 0 \quad (2.64)$$

$$\Leftrightarrow \langle MT_m - T_{\hat{k}}, [(1 - 2w_m^c(x))M + 2w_m^c(x)I_e]T_m + T_{\hat{k}} - 2u(x) \rangle > 0. \quad (2.65)$$

Equation (2.65) is similar to eq. (2.53). Therefore, for three classes or more, we can find a displacement of the target  $T_m$  similar to eq. (2.54) which breaks inequality (2.65), namely

$$MT_m = -T_m - \frac{T_{\hat{k}} - u(x)}{w_m^c(x)}. \quad (2.66)$$

See Claim 8 for the two-class case.  $\square$

**Claim 8.** *With two classes, inequality (2.40) holds for any target displacement matrix  $M$  for samples not belonging to the class of the displaced target ( $\hat{k} \neq m$ ).*

*Proof.* Let us set  $\hat{k} = 1$  and  $m = 2$ . Similarly to the developments made in section 2.4.3, we have

$$(2.65) \Leftrightarrow (1 - 2w_2^c(x))|MT_2 - T_1|^2 > 0. \quad (2.67)$$

Since the sample  $x$  belongs to class 1, then  $w_2^c(x) < 1/2$ , and therefore  $1 - 2w_2^c(x) > 0$ . So eq. (2.67) indeed holds for any  $M$ .  $\square$

## 2.5 Optimal Targets for 3 Classes or More

### 2.5.1 Variational Formulation

Looking for optimal targets could be done by minimizing the empirical risk, which is a complex task in a multi-class context Zhang [2004]. A simpler approach is to instead minimize a cost function comparing  $u_i$  with  $t_i$ ,  $i \in [1..n]$ , similar to a regression loss, but with the  $T_k$ 's,  $k \in [1..p]$ , as unknowns as opposed to some parameters that would define  $u_i$  while using fixed  $T_k$ 's. We propose to use definition 2.3.

**Definition 2.3. Cost function.** We define the cost function  $F$  as

$$F(T_1, \dots, T_p) = \sum_{i=1}^n |u_i - t_i|^2. \quad (2.68)$$

The intuition behind definition 2.3 is that targets producing a low value of  $F$  should also ensure good classification performances. Note that, as already mentioned, minimizing a regression loss in hope that it would work for classification is also the principle of classification neural network optimization. If regarding the optimization problem as dealing with some parameters and the targets altogether, the difference is that, here, the parameters are predefined and fixed while, with neural networks, it is the case of the targets.

Since  $u_i$  is a weighted sum of the targets, a trivial way to reach the lowest possible value of  $F$  is to set all the targets at the origin. Hence, to find useful targets, some constraints must be added. One way is to impose a lower bound on the distance between any two targets

$$\forall k < l, |T_k - T_l|^2 \geq \delta^2 \quad (2.69)$$

where  $\delta$  is a positive constant.

**Definition 2.4. Constraint on Targets.**  $\mathcal{E}_\delta$  denotes the target constraint domain defined by eq. (2.69), and  $\partial\mathcal{E}_\delta$  denotes its boundary.

**Conjecture 2.1.**  $\partial\mathcal{E}_\delta$  contains the regular  $(p-1)$ -simplices of side length  $\delta$ .

**Claim 9.**  $F$  is a quadratic form in  $U$  defined by a real, symmetric matrix  $Q$  of the form  $\text{Diag}_e(Q^\square)$ .

*Proof.* We have

$$u_i = \sum_{k=1}^p w_k^c(i) T_k \quad (2.70)$$

$$= \text{Vec}_{j=1}^e \left( \begin{matrix} [w_1^c(i) \ w_2^c(i) \ \dots \ w_p^c(i)] & \begin{bmatrix} T_1[j] \\ T_2[j] \\ \vdots \\ T_p[j] \end{bmatrix} \end{matrix} \right) \quad (2.71)$$

$$= \text{Vec}_{j=1}^e \left( (C\Omega^s e_i^n)^\top U_j \right) \quad (2.72)$$

$$= \text{Diag}_e((C\Omega^s e_i^n)^\top) U \quad (2.73)$$

$$= A_i U. \quad (2.74)$$

We also have

$$T_k = \text{Diag}_e(e_k^{p\top})U = B_k U. \quad (2.75)$$

Then,

$$F(T_1, \dots, T_p) = \sum_{i=1}^n |u_i - t_i|^2 \quad (2.76)$$

$$= \sum_{k=1}^p \sum_{\substack{i \\ t_i=T_k}} |A_i U - T_k|^2 \quad (2.77)$$

$$= \sum_{k=1}^p \sum_{\substack{i \\ t_i=T_k}} |(A_i - B_k)U|^2 \quad (2.78)$$

$$= \sum_{k=1}^p \sum_{\substack{i \\ t_i=T_k}} ((A_i - B_k)U)^\top (A_i - B_k)U \quad (2.79)$$

$$= U^\top \sum_{k=1}^p \sum_{\substack{i \\ t_i=T_k}} (A_i - B_k)^\top (A_i - B_k)U \quad (2.80)$$

$$= U^\top Q U. \quad (2.81)$$

appendix A.3 shows that the matrix  $Q$  is equal to  $\text{Diag}_e(Q^\square)$  with

$$Q^\square = C(\Omega^s - I_n)(\Omega^{s\top} - I_n)C^\top. \quad (2.82)$$

Clearly,  $Q^\square$  is symmetric, and so is  $Q$  then.

Note that this expression of  $Q^\square$  is not used here. However, it is provided for completeness and might be useful in some future developments of the proposed method.  $\square$

## 2.5.2 Minimization of $F$

**Definition 2.5. Optimal Targets.** A target set is called optimal if and only if it minimizes  $F$  while belonging to  $\mathcal{E}_\delta$ .

**Claim 10.**  $F$  is constant on the set of regular simplices with side length  $\delta$ . Its value has the form  $\alpha\delta^2$  where  $\alpha$  is a positive real number depending only on the learning sample weights  $w_j^s(i)$ ,  $(i, j) \in [1..n] \times [1..p]$ , and the class assignments of the learning samples.

*Proof.*

$$F(T_1, \dots, T_p) = \sum_{k=1}^p \sum_{\substack{i \\ t_i=T_k}} \left| \sum_{l=1}^p w_l^c(i) T_l - T_k \right|^2 \quad (2.83)$$

$$= \sum_{k=1}^p \sum_{\substack{i \\ t_i=T_k}} \underbrace{\left| \sum_{l=1}^p w_l^c(i) (T_l - T_k) \right|^2}_{\mathcal{A}_{i,k}}, \quad (2.84)$$

where

$$\begin{aligned} \mathcal{A}_{i,k} &= \sum_{l=1}^p (w_l^c(i))^2 |T_l - T_k|^2 \\ &\quad + 2 \sum_{l < m} w_l^c(i) w_m^c(i) (T_l - T_k) \cdot (T_m - T_k). \end{aligned} \quad (2.85)$$

If the target set belongs to  $\partial\mathcal{E}_\delta$ , then  $|T_l - T_k|^2$  is equal to  $\delta^2$  if  $l \neq k$ , by definition, and

$$(T_l - T_k) \cdot (T_m - T_k) = \begin{cases} \delta^2 \cos(\angle_k^{l,m}) & \text{if } l \neq k \text{ and } m \neq k, \\ 0 & \text{otherwise} \end{cases}, \quad (2.86)$$

where  $\angle_k^{l,m}$  is the angle at  $T_k$  formed with  $T_l$  and  $T_m$ , and is actually independent of  $k, l$ , and  $m$ , and equal to  $\pi/3$  since the target set forms a regular simplex. Therefore (see Appendix A.2),

$$\mathcal{A}_{i,k} = \frac{\delta^2}{2} \left( \sum_{l=1}^p (w_l^c(i))^2 - 2w_k^c(i) + 1 \right), \quad (2.87)$$

and (see appendix A.4)

$$F(T_1, \dots, T_p) = \frac{\delta^2}{2} (|\Omega^c|_F^2 - 2\text{Tr}(\Omega^c C^\top) + n), \quad (2.88)$$

where  $|M|_F$  is the Frobenius norm of  $M$ . Clearly, the constant multiplying  $\delta^2/2$  in eq. (2.88), say  $\alpha$ , cannot be negative since  $F$  is non-negative. Let us show that  $\alpha$  is also not equal to zero if  $w_\gamma$  has an infinite support. We have

$$\alpha = \sum_{k=1}^p \left( \sum_{i=1}^n (w_k^c(i))^2 - 2 \sum_{\substack{i \\ t_i=T_k}} w_k^c(i) + \sum_{\substack{i \\ t_i=T_k}} 1 \right) \quad (2.89)$$

$$= \sum_{k=1}^p \left( \sum_{\substack{i \\ t_i \neq T_k}} (w_k^c(i))^2 + \sum_{\substack{i \\ t_i=T_k}} \left( (w_k^c(i))^2 - 2w_k^c(i) + 1 \right) \right) \quad (2.90)$$

$$= \sum_{k=1}^p \left( \sum_{\substack{i \\ t_i \neq T_k}} (w_k^c(i))^2 + \sum_{\substack{i \\ t_i=T_k}} (w_k^c(i) - 1)^2 \right). \quad (2.91)$$

Because  $\alpha$  is a sum of non-negative terms, it can be equal to zero if and only if all the terms are equal to zero. According to section 2.3.3, this cannot happen.  $\square$

**Conjecture 2.2. Optimal Targets.** *Target sets representing regular simplices with side-length  $\delta$  are optimal in the sense of definition 2.5. Any such target set can be selected as a solution for an arbitrary value of  $\delta$ . Two ‘interesting’ values of  $\delta$  are 1 and  $\sqrt{2}$ . If choosing  $\sqrt{2}$ , then an optimal target set can be easily defined as follows*

$$T_k = e_k^{p-1}, k \in [1..p-1], \quad (2.92)$$

$$T_p = \frac{\sqrt{p} + 1}{p-1} \mathbf{1}_{p-1}. \quad (2.93)$$

**Theorem 2.1. Equivalence to wNN Classifier.** *When choosing the targets optimally (see Conjecture 2.2), then the proposed classifier (see definition 2.1) corresponds to a weighted Weighed-Nearest-Neighbours (WNN) classifier.*

*Proof.* Note that the expression (2.87) of  $A_{i,k}$  for the sample  $x_i$  is actually valid for any sample  $x$ . Let us rename it  $A_k(x)$  in this case. Then, the Nearest Target (NT) classifier can be rewritten as follows

$$\hat{k}(x) = \arg \min_{k \in [1..p]} |u(x) - T_k|^2 \quad (2.94)$$

$$= \arg \min_{k \in [1..p]} A_k(x) \quad (2.95)$$

$$= \arg \min_{k \in [1..p]} \frac{\delta^2}{2} \left( \sum_{l=1}^p (w_l^c(x))^2 - 2w_k^c(x) + 1 \right) \quad (2.96)$$

$$= \arg \max_{k \in [1..p]} w_k^c(x). \quad (2.97)$$

□

### 2.5.3 Classification Point-of-View

The optimality condition of Conjecture 2.2 is related to the cost function (2.68) defined as a tractable alternative to the empirical risk. To support the idea that choosing the targets equidistant from each other is, if not provably optimal, also a good choice in terms of classification performances, we propose to study these performances in terms of how the target set moves away from equidistance. To allow for the existence of a regular polytope with triangular faces formed by the targets, we must have  $e \geq p - 1$ . Then, equidistance can be replaced with equality of the angles formed by any two tangent edges of the target polytope. From this point of view, aligned targets (which corresponds to degenerate, flat triangular faces) can be considered as being as far as possible from equidistance. Claim 11 states what happens in this case.

**Claim 11.** *The degenerate case where the targets are aligned can be viewed as being as far as possible from the optimality condition of Conjecture 2.2. In that case, the proposed classifier only predicts two out of the  $p$  classes, which indeed severely impairs its performances.*

*Proof.* Let the targets  $T_k$ ,  $k \in [1..p]$ , be distinct and aligned. Since the classifier is invariant to global translation and rotation of the targets (this can be easily verified from eq. (2.9)), it can be assumed without loss of generality that

$$T_1 \neq \mathbf{0}_e, \quad (2.98)$$

$$\forall k \in [1..p], T_k = \alpha_k T_1, \quad (2.99)$$

$$1 = \alpha_1 < \alpha_2 < \dots < \alpha_p. \quad (2.100)$$

Then,

$$u(x) - T_k = \sum_{l=1}^p w_l^c(x) T_l - \sum_{l=1}^p w_l^c(x) T_k \quad (2.101)$$

$$= \sum_{l=1}^p w_l^c(x) (T_l - T_k) \quad (2.102)$$

$$= T_1 \sum_{l=1}^p w_l^c(x) (\alpha_l - \alpha_k) \quad (2.103)$$

$$= T_1 \sum_{l=1}^p w_l^c(x) \alpha_l - \alpha_k. \quad (2.104)$$

Therefore,

$$\text{eq. (2.9)} \Leftrightarrow \hat{k}(x) = \arg \min_{k \in [1..p]} \left| \sum_{l=1}^p w_l^c(x) \alpha_l - \alpha_k \right| \quad (2.105)$$

$$\Leftrightarrow \hat{k}(x) = \begin{cases} 1 & \text{if } \sum_{l=1}^p w_l^c(x) \alpha_l \leq (\alpha_1 + \alpha_p)/2 \\ p & \text{otherwise} \end{cases}. \quad (2.106)$$

Note that eq. (2.106) should actually distinguish the cases ‘strictly lower’, ‘strictly higher’, and ‘equal’, where equality requires to take an arbitrary decision between classes 1 and  $p$ . Nevertheless, it is still true that the classifier can only predict two classes out of  $p$ .  $\square$

Keeping the triangular faces’ regularity point-of-view of equidistance, the target set regularity can be characterized by an appropriate function of the angles between any two tangent edges, typically minimal for a regular polytope. For example, if  $p = 3$  (the target polytope is a triangle), the absolute difference between the extrema of the angles can be used. It ranges from zero (equidistance) to  $\pi$  (aligned targets). Let us see how the classification performances vary as a function of this quantity in the following experiment, see fig. 2.3. We observe that the lowest values of the empirical risk are obtained for low differences between the extreme angles, *i.e.*, for equidistant targets and configurations close to it. This is in accordance with the Conjecture 2.2. Details on the experiment and additional figures are given in appendix C.1 .

## 2.6 Selection of $\gamma$

In this section, we consider the case where all  $\gamma_j, j \in [1..n]$ , are equal to a unique value  $\gamma$  and the proportions of learning sample per class are the same.

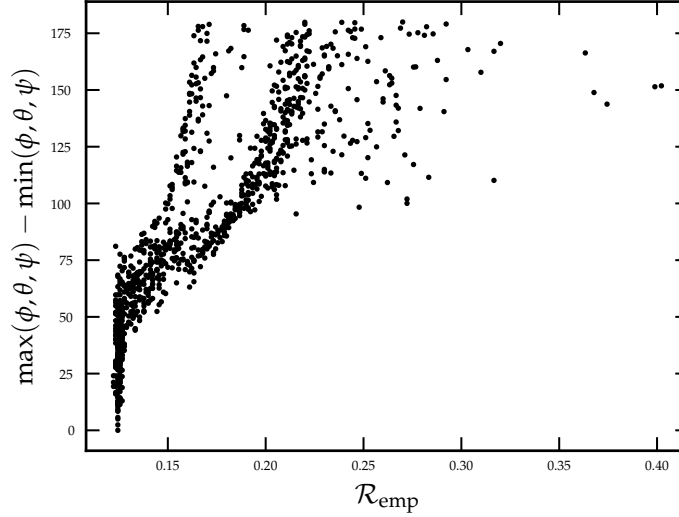
### 2.6.1 Limit Cases

The sample weights  $w_j^s(x), j \in [1..n]$ , implicitly depend on  $\gamma$  (see eq. (2.13)). Let us check their limit values when  $\gamma$  tends toward zero or infinity, and the consequences on the proposed classifier.

**Claim 12.** *When  $\gamma$  tends toward zero, the proposed classifier becomes unusable.*

*Proof.* Let  $x$  be a sample to classify. From eq. (2.13), we have

$$\lim_{\gamma \rightarrow 0} w_j^s(x) = \gamma \alpha(x) w^{\text{rad}}(0), \quad (2.107)$$



**Figure 2.3** Empirical risk of classification into 3 classes as a function of the absolute difference between the extrema of the target triangle angles ( $\phi, \theta, \psi$ , in degrees). Each dot represents an iteration (among 1000) of the classification of  $m = 900$  test samples (based on other  $n = 900$ ) with targets drawn from a random and uniform distribution (in the range  $[0,1]$ ). The parameters  $\gamma_j$  are all set to 100. The equidistant targets corresponds to a null angle difference. See appendix C.1 for details and additional figures.

from which it can be checked that

$$\lim_{\gamma \rightarrow 0} w_j^s(x) = 1/n \quad (2.108)$$

since the weights must sum to one. Then,

$$\forall k \in [1..p], w_k^c(x) = \frac{1}{n} \text{Card}(\{i | t_i = T_k\}) \quad (2.109)$$

and

$$u(x) = \frac{1}{n} \sum_{k=1}^p \text{Card}(\{i | t_i = T_k\}) T_k. \quad (2.110)$$

All the samples being transformed into a common point, it is of course not possible to take a classification decision.  $\square$

**Theorem 2.2. Equivalence to Nearest-Neighbour Classifier.** *When  $\gamma$  tends toward infinity, the learning samples are correctly classified. For samples not in the learning set, the behaviour depends on the weighting function. For a Gaussian, the limit classifier is the Nearest-Neighbour (NN) classifier. appendix A.6 analyzes another example of weighting function.*

*Proof.* It follows from section 2.3.3 that, when  $\gamma$  tends toward infinity, the learning samples are transformed into their corresponding target. Consequently, they are correctly classified.

For a sample  $x$  not in the learning set, the limit weights depend on the weighting function. If the weighting function is a Gaussian, then

$$w^{\text{rad}}(\gamma | x - x_i |) = \alpha_\gamma e^{-\gamma^2 | x - x_i |^2}, \quad (2.111)$$



where  $\alpha_\gamma$  is a normalization constant. Then, for  $i \neq j$ ,

$$\lim_{\gamma \rightarrow +\infty} \frac{w^{\text{rad}}(\gamma|x - x_i|)}{w^{\text{rad}}(\gamma|x - x_j|)} = \lim_{\gamma \rightarrow +\infty} e^{\gamma^2(|x-x_j|^2 - |x-x_i|^2)} \quad (2.112)$$

$$= \begin{cases} +\infty & \text{if } |x - x_i| < |x - x_j| \\ 1 & \text{if } |x - x_i| = |x - x_j| \\ 0 & \text{otherwise} \end{cases}. \quad (2.113)$$

Therefore,

$$\lim_{\gamma \rightarrow +\infty} w_j^s(x) = \begin{cases} 1 & \text{if } \forall i \neq j, |x - x_j| < |x - x_i| \\ 0 & \text{if } \exists i \neq j, |x - x_j| > |x - x_i|, \\ \frac{1}{1+q} & \text{otherwise} \end{cases}, \quad (2.114)$$

where  $q = \text{Card}\{i \neq j \mid |x - x_i| = |x - x_j|\}$ . In terms of classification, eq. (2.114) means that if  $x$  is (strictly) inside the voronoi cell of a learning sample, it will be assigned the same class as this sample (weight 1 for it, zero for the others), and if  $x$  is on a voronoi frontier, it will be assigned the most present class among the tangent voronoi cells, or not classified in case of a tie (weights  $1/(1+q)$  for the  $q$  tangent cells). Overall (i.e., samples not in the learning set and learning samples), this corresponds to the Nearest-Neighbor classifier.  $\square$

### 2.6.2 About an Optimal Value

We have conjectured an optimality condition on the targets using the cost function of definition 2.3. With such targets, we can now view eq. (2.68) as a function of  $\gamma$  to look for an optimal value. Let us write

$$F(\gamma) = \sum_{i=1}^n |u_i - t_i|^2 \quad (2.115)$$

$$= \sum_{i=1}^n \left| \sum_{j=1}^n w_j^s(i)(t_j - t_i) \right|^2, \quad (2.116)$$

where the dependence on  $\gamma$  is hidden in  $w_j^s(i)$ .

**Claim 13.** *F, seen as a function of  $\gamma$ , cannot be used to find an optimal value of  $\gamma$ .*

*Proof.* Due to the similitude between eq. (2.116) and eq. (2.84), we can use eq. (2.87) with the replacements

$$\sum_{l=1}^p \longrightarrow \sum_{l=1}^n, \quad (2.117)$$

$$w_l^c \longrightarrow w_l^s, \quad (2.118)$$

$$T_l \longrightarrow t_l, \text{ and } T_k \longrightarrow t_i, \quad (2.119)$$

to get

$$F(\gamma) = \frac{\delta^2}{2} \sum_{i=1}^n \left( \sum_{j=1}^n (w_j^s(i))^2 - 2w_i^s(i) + 1 \right) \quad (2.120)$$

$$= \frac{\delta^2}{2} \sum_{i=1}^n \left( \sum_{\substack{j=1 \\ j \neq i}}^n (w_j^s(i))^2 + (w_i^s(i) - 1)^2 \right). \quad (2.121)$$

As a sum of non-negative terms, the expression (2.121) reaches its minimal value of zero only if all the terms  $w_j^s(i), j \neq i$  and  $w_i^s(i) - 1$  are equal to zero. According to section 2.3.3, this cannot happen for finite values of  $\gamma$ . Consequently,  $F$  cannot be used to find an optimal value of  $\gamma$ .  $\square$

Instead of finding an optimal value for  $\gamma$  analytically, an alternative is to use CV.

## 2.7 Some Choices of parameters $\gamma_i, i \in [1..n]$

Instead of considering a unique parameter  $\gamma$ , one can use a parameter per learning samples  $\gamma_i, i \in [1..n]$ . Exploring the parameter space by CV would be unfeasible for large data sets (goes as  $m^n$  for  $m$  parameters to explore). Instead, one can inspire from existing methods, that also relies on defining similarities between samples, and in particular, non-linear Dimension Reduction (DR) methods. A popular method is t-SNE [Van der Maaten and Hinton, 2008], It aims at measuring significant similarities between samples that live in *high*-dimensional spaces, in order to represent them in a *lower*-dimensional space, as good as possible *i.e.*, based on an optimal criterion. It takes into account the local density of samples to define weights between them. It uses almost the same definition<sup>3</sup> of the weights  $w_i^s$ s between the samples as in definition 2.2, with the parameters  $\sigma_i, i \in [1..n]$  such that  $\gamma_i = 1/(2\sigma_i^2)$ . The authors propose to adapt the parameters  $\sigma_i$  to the local density, which is also relevant for our case (see an example in fig. C.5). Indeed, it is more appropriate to take a smaller value for  $\sigma_i$  (inverse for  $\gamma_i$ ) in the denser regions. Hence, the samples-to-samples weights will not be biased toward the denser regions. More recently, other DR methods have been proposed. For example, UMAP (Uniform Manifold Approximation and Projection) [McInnes et al., 2018] is also popular. It also relies on the definition of per-sample parameters  $\gamma_i$ s, which is different, but remains similar in the philosophy. We do not give more details here since these definitions are not used in practice. Instead, we take the advantage of the equivalence of the proposed method to the WNN (for equidistant targets). Then, we take into account the sample local density with a definition of the weights  $w_i^s$ s that rely on the neighbours order. This definition is motivated by its property of optimality for the classification with a WNN (see details in section 3.2.1).

---

<sup>3</sup>The difference comes from their definition of the weight of a sample to itself, that is set to zero.



## Chapter 3

# Implementation and Experimental Results

### **Key points – The method in practice**

1. We give details on a specific implementation of the  $W$ - $k$ -NN, designed to handle modern image data sets with GPU support.
2. We show the efficiency of the proposed implementation.

### **Contributions – Details of the implementation**

3. Definition of the sample weights used for the applications.
4. Procedure to search for the best parameter using  $CV_{LOO}$  estimations.
5. Experimental results on a 2-d synthetic data set.
6. Experimental results on CIFAR-10.

**Chapter 3 – Implementation and Experimental Results:**

3.1	Purpose . . . . .	34
3.2	The method in practice . . . . .	34
3.2.1	Asymptotically optimal weights . . . . .	34
3.2.2	Leave-One-Out Cross-Validation . . . . .	35
3.2.3	Per-class weights for unbalanced data sets . . . . .	35
3.2.4	Numerical implementation with KeOps . . . . .	36
3.3	Experimental results on synthetic data . . . . .	36
3.3.1	Comparison to standard methods . . . . .	36
3.3.2	Procedure to use the NN-Kernel . . . . .	38
3.4	Experimental results on CIFAR-10 . . . . .	38

**3.1 Purpose**

The literature contains a vast catalogue of supervised machine learning methods, with impressive results on benchmark data sets, it is often time-consuming to dig into it and pick the right method adapted to the desired application. Indeed, State-Of-The-Art methods are designed to perform well in specific areas and according to certain metrics (*e.g.*, accuracy, time, memory, *etc*) but rarely take into account the interpretability of the predictions while it is often needed for real applications. In chapter 2, we proposed a new interpretation of the WNN classifier. Here, we propose a modern implementation of the WNN classifier. It features the definition of the optimal weights from Samworth [2012] that relies on only one, interpretable, parameter: the number of neighbours to consider. As we will see in chapter 4, this classifier can perform very well on plankton image data sets. Therefore, we developed a modern implementation using Graphics Processing Unit (GPU) computing. Hence, the method is easy to use in practice and interpretable.

This chapter aims at presenting the implementation in detail and showing its ability to deal with modern, large data sets. Results on real plankton images data sets are left for the chapter 4.

**3.2 The method in practice**

This section gives details on the implementation used for the experiments presented here and in chapter 4. The implementation using *python* is available on the Inria GitLab<sup>1</sup>.

**3.2.1 Asymptotically optimal weights**

The proposed classifier results in a WNN for the optimal choice of targets (see Theorem 2.1). The model relies on the choice of the weights' definition (in particular, the function  $w^{rad}$  in definition 2.2). A standard choice for  $w^{rad}$  is a Gaussian with parameters  $\gamma_i = 1/(2\sigma_i^2)$ ,  $i \in [1..n]$ , but this choice is subjective and may be limiting in practice. Instead, we propose to use a result from Samworth [2012] that defines *asymptotically optimal weights* for the WNN. To be exact, those are weights that are asymptotically (*i.e.*,  $n \rightarrow \infty$ ) optimal in the sense of minimizing the risk<sup>2</sup> (eq. (2.3)). Let us denote such weights as '*optimal weights*'.

In theory, the definition of the weights does not rely on a parameter. Nevertheless, in practice, the authors propose to tune a unique parameter,  $k$ : the number of neighbours to consider. Hence, we denote this specific implementation of the WNN as the Weighed- $k$ -Nearest-Neighbours (W- $k$ -NN). The weights are a function of the neighbours' *rank* instead of the distance. Here the rank

<sup>1</sup><https://gitlab.inria.fr/cedubois/w-k-nn/>

<sup>2</sup>see Theorem 2 in Samworth [2012]

$r$  denotes the index of the list of the  $k$  neighbours ordered in increasing order *i.e.*,  $r \in \{1 \dots k\}$ . The weight function between two samples is

$$w^*(x, x') = \begin{cases} \frac{1}{k^*} \left[ 1 + \frac{d}{2} - \frac{d}{2k^{*2/d}} (r^{1+2/d} - (r-1)^{1+2/d}) \right] & \forall r \leq k^* \\ 0 & \text{otherwise} \end{cases}, \quad (3.1)$$

where  $r$  is the rank of the sample  $x'$ ,  $d$  the dimension of the sample space and  $k^*$  the number of neighbours to consider, to be determined via CV. All possible values of  $k$  can be tested (from one to  $n$ ), which guarantees to provide *the* optimal  $k^*$  among  $n$  for the data at hand. Nevertheless, in practice, we do not expect a large value of  $k^*$  since the samples should be ‘grouped’ in the feature space (at least for a relevant one).

### 3.2.2 Leave-One-Out Cross-Validation

We propose to search for the optimal number of neighbours  $k^*$  through the minimization of the  $CV_{LOO}$  risk (see section 2.1.2) for a given set of  $\{k_i | 0 < k_1 < k_2 \dots k_l \leq n\}_{i=1}^l$ . In order to compute efficiently (*i.e.*, in reasonable time) a  $CV_{LOO}$  score, we compute the  $CV_{LOO}$  transformation for the learning samples, omitting the contribution of the sample to itself

$$u^{\{-j\}}(x_j) = \sum_{i \neq j}^n w_i^s t_i, \quad (3.2)$$

to then perform the NT classification. Note that, for the sake of completeness, we keep the notation of the targets general (*i.e.*, not restricted to the canonical basis).

In practice, to compute eq. (3.1), we first need to compute the search for the  $k$  neighbours. Then we store the  $n \times k_{\max}$  rank matrix to re-use them for each prediction with  $k < k_{\max}$ . Hence, the nearest-neighbours search is computed only once for  $k_{\max}$ .

Gathering everything together, the final implementation allows performing experiment on real data-sets without the need of an expertise to tune the method. Indeed, the only parameter is the number of neighbours to consider, and it is automatically set by minimizing the  $CV_{LOO}$  risk.

### 3.2.3 Per-class weights for unbalanced data sets

Real data sets often present imbalance between classes, *i.e.*, the number of samples per class is not identical among classes. This is limiting, since most of the methods are generally not designed for that situation. Nevertheless, a common usage in practice is to define weights according to the class frequency (*i.e.*, number of occurrences per class). This allows reducing the tendency of the methods to perform better on the most represented classes. We propose to use this technique, and we present its implication for the proposed classifier (including W- $k$ -NN). Let us define the ‘scaled’ transformation as

$$u(x) = \sum_{j=1}^p \eta_j w_j^c t_j, \quad (3.3)$$

with

$$\eta_j = \frac{\alpha}{|y_j|}, \quad (3.4)$$

where  $|\cdot|$  is the cardinal, such that  $|y_i|$  is the number of elements in the class ‘ $i$ ’ and with the normalization  $\alpha = 1 / \sum_{j=1}^p \frac{w_j^c}{|y_j|}$ , such that  $\sum_j^p \eta_j w_j^c = 1$ . In terms of classification, by considering

the NT with optimal targets ( $W$ - $k$ -NN), the predicted class is the one with the highest new class-weight  $\eta_j w_j^c$  (see Theorem 2.1). The impact of the scaling is a bias toward the less represented classes. This bias is inversely proportional to the class frequency; this choice is subjective.

With the normalization, the ‘scaled’ transformation still corresponds to the definition 2.1.

### 3.2.4 Numerical implementation with KeOps

Distance-based methods such as  $k$ -NN are computationally intensive since they rely on calculating sample-to-sample distances. This limits their applications, specially for modern, large scale data sets. Strategies have been proposed to reduce the computation cost. For example, tree-based search methods have demonstrated high acceleration [Bentley, 1975; Jiang et al., 2017]. Another possibility is to use the approximate nearest neighbour search methods (e.g., Wang et al. [2021]), that could be efficient for the  $W$ - $k$ -NN.

We propose to use KeOps [Charlier et al., 2021], a recent library designed to compute fast *Kernel Operations* on GPUs. Hence, the computation of the nearest neighbours search is efficient. Note that we use the Euclidean distance to compute the search of the nearest-neighbours in all our experiments.

Finally, our experiments were all performed on the same computer, a Dell workstation 7740 (2020) with an Intel<sup>®</sup> Xeon<sup>®</sup> E-2286M CPU @ 2.40 GHz, an Nvidia Quadro RTX 5000 GPU and 64 Go of Random Access Memory.

## 3.3 Experimental results on synthetic data

### 3.3.1 Comparison to standard methods

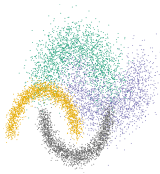
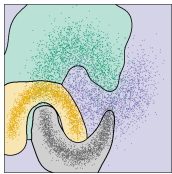
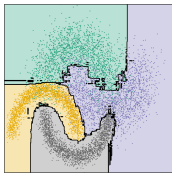
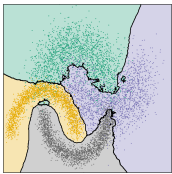
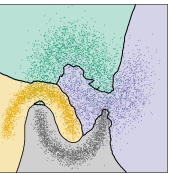
In this section we consider a synthetic data set of  $n = 8,000$  samples,  $d = 2$  dimensions and  $p = 4$  classes. Samples are represented on the first column of table 3.1. This defines a multi-class and non-linear classification situation. We present classification results with the proposed implementation for the  $k$ -NN and  $W$ - $k$ -NN classifiers, and compare it with two other standard methods, a SVM with a Gaussian kernel and a RF. Additionally, we detail the results of a SVM with the Nearest-Neighbour-Kernel (NN-Kernel) in section 3.3.2.

We used a CV strategy to optimize the parameters of each method, based on the ACC. For the  $k$ -NN and  $W$ - $k$ -NN we used the  $CV_{LOO}$  while for the RF and SVM we used a 10-folds CV since the computation of the  $CV_{LOO}$  would be time-consuming (due to the fit for each fold). For the SVM, there are two parameters to optimize, the regularization  $C$  and the scale parameter  $\gamma$  of the kernel (here  $\gamma = 1/(2\sigma^2)$ ). The ‘one-versus-one’ strategy is used for the multi-class setting. We tested 3 values of  $C$  (1, 10, 100) and 3 values of  $\gamma$  (1, 10, 100), to cover a wide range. The set of parameters that produces the best score over the 10 folds is ( $C = 10$ ,  $\gamma = 100$ ). For the RF, the only parameter tested is the maximum depth of the trees, among (3, 5, 10, 15). The forest is composed of 100 trees and the minimum numbers of samples per leaf is set to one. The best score is obtained for a maximum depth of 10. For the  $W$ - $k$ -NN, we used the optimal weights given in section 3.2.1, the parameter to set is  $k$ , as for the  $k$ -NN. Let us remind that the rank matrix is stored, after it is computed for  $k_{\max}$ , and re-used for  $k < k_{\max}$  (see section 3.2.2). Hence, we were able to test  $n_k = 20$  values of  $k$  for both methods with a  $CV_{LOO}$  for each, in a relatively short amount of time, compare to a standard implementation. The optimal values are  $k = 20$  for  $W$ - $k$ -NN and  $k = 30$  for  $k$ -NN. The overall results are summarized in table 3.1, with an overview of the decision frontiers. Note, all the methods are competitive as they have similar accuracy scores (the deviation among methods is  $<1\%$ ).

The implementations of the  $k$ -NN and the  $W$ - $k$ -NN are based on KeOps , which takes advantage of the GPU support for fast computation. In order to get comparable results, for the SVM we also used an implementation with a GPU support: ThunderSVM [Wen et al., 2018]. For the RF method, we tested the XGBoost implementation [Chen and Guestrin, 2016] (that has GPU support to estimate the best splits in the tree construction), but the computation time was equivalent or worse than the standard scikit-learn implementation [Pedregosa et al., 2011] (in that specific case). We choose to use the latter for simplicity. Note, we were not able to use the ThunderGBM implementation [Wen et al., 2020] for tree-based methods. The computation times for the fits and the predictions (normalized to the  $W$ - $k$ -NN ones) are given on table 3.1.

One major observation (see table 3.1) that can be safely state is that  $k$ -NN is faster than  $W$ - $k$ -NN (the version with optimal weights). This is explained by the additional operations needed for the  $W$ - $k$ -NN. Indeed, the  $k$ -NN prediction can be computed by taking the most present label among of the neighbourhood samples, while, for its weighted version, the weights have to be computed and a sum over the neighbours is necessary. While, in this toy example, the ACC is the same for both, it may differ for other applications (see Samworth [2012]). With the SVM classifier, the fitting time depends on the optimization (gradient descent here) but is generally expected to be time-consuming. On the other hand, the prediction time is comparable with the others, which is interesting for real applications. Then, concerning the RF, the optimization is faster than for the SVM, probably because of the simplicity of the model (splits). But on the other hand, the prediction is slower (almost by a factor 3). These observations have to be mitigated with the setting of the experience, and in particular the implementations of the methods.

As an additional test, we computed (i) the  $CV_{LOO}$  of the  $k$ -NN with the standard scikit-learn implementation (brute force search on CPU) for all the 20 values of  $k$  and observed that the fitting procedure is 40 times slower than ours and is 35 times slower for the prediction. The gain is mainly coming from the GPU support from the KeOps library, even only for the prediction. Our main contribution here comes from the proposition for computing the  $CV_{LOO}$  (in section 3.2.2).

	Gaussian-SVM (ThunderSVM)	RF (scikit-learn)	$k$ -NN (KeOps)	$W$ - $k$ -NN (KeOps)
				
CV	10-folds	10-folds	$CV_{LOO}$	$CV_{LOO}$
Grid search size	$3 \times 3$	4	20	20
Optimal params.	$C=10, \gamma=100$	max depth = 10	$k = 30$	$k = 20$
CV fitting time	48.33	3.88	0.88	1
Prediction time	1.26	3.47	0.28	1
ACC (%)	90.6	90.2	90.5	90.5

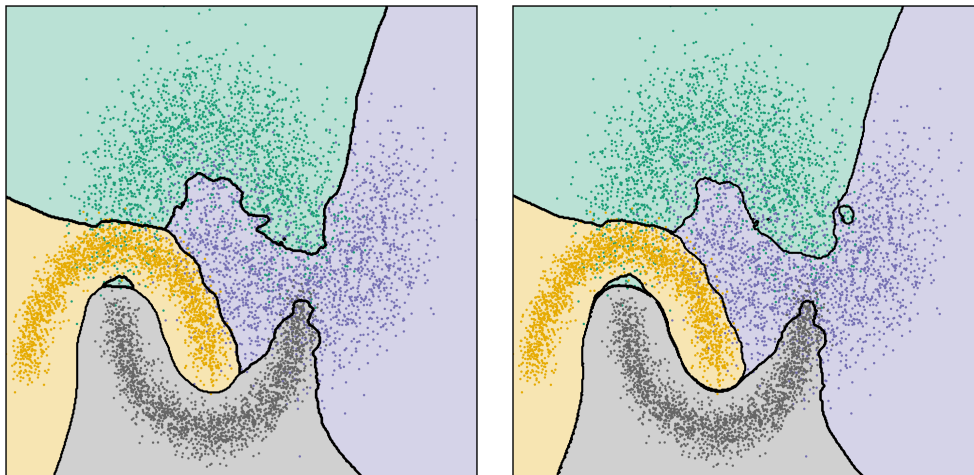
**Table 3.1** Classification results for the synthetic data set. The first image show the samples and their label as colour. Those after show the same samples and the decision frontiers plotted in black from predictions on a fine, regular sample grid. The last six rows give details on the CV procedure used to estimate the optimal parameters (see text for details). The duration of the fitting and the prediction are normalized by the ones from the  $W$ - $k$ -NN method to improve the reading. For the prediction, it is repeated 1000 times to get reliable estimations (since typical values are  $<0.1s$ ).



### 3.3.2 Procedure to use the NN-Kernel

In practice, the NN-Kernel (see definition in Claim 2) can be used with a linear classifier (another than the NT) to improve the results of the  $W$ - $k$ -NN method (that defines the so-called *kernel trick*). This amounts to transforming the samples to the target space of dimension  $p$  ( $p - 1$  or more to be exact, see section 2.5) and fitting a linear classifier in that space. The standard linear classifiers that take advantage of the kernel trick is the SVM classifier, that we use here.

First we propose to estimate  $k^*$ , the optimal parameter, of the transformation via  $CV_{LOO}$  with the NT classifier and then to optimize the regularization parameter  $C$  of the linear SVM on these transformed samples, this time with a  $K$ -fold CV to accelerate the process. This result in a ‘hybrid’  $CV_{LOO}$  -  $K$ -folds CV procedure to optimize the parameters ( $k^*$  and  $C$ ). An exact  $CV_{LOO}$  estimation would be very time-consuming (all combinations of  $k^*$  and  $C$  for  $n$  folds). Still, this pipeline is expected to be reliable regarding the results obtained with the NT classifier only. As an example, let compute the NN-Kernel-SVM prediction for the previous synthetic data set. The CV of the linear SVM increases the total fitting time from 0.9s (for  $W$ - $k$ -NN) to 97.5s (almost  $\times 110$ ). For the prediction, it increases by a factor 2.3 (from 5.8 for the  $W$ - $k$ -NN to 13.1). The classes-frontier is plotted on fig. 3.1; the mean ACC over 10-folds is 90.7 (similar with the Gaussian-SVM, see table 3.1). This application demonstrates that the NN-Kernel can be used with a SVM classifier in practice. Nevertheless, there may not be a practical benefit compared to using a  $W$ - $k$ -NN or a kernel-SVM given the fitting and prediction time it requires.



**Figure 3.1** Decision boundaries for the  $W$ - $k$ -NN (left) and NN-Kernel-SVM (right) classifiers over the synthetic data set (see section 3.3.1). The coloured dots are the learning samples, with a colour per label.

## 3.4 Experimental results on CIFAR-10

We ran a set of experiments on the data set CIFAR-10 [Krizhevsky, 2009]. It is composed of  $n = 60,000$  colour images among 10 classes. In order to extract some features of the images, we rely on a pre-trained CNN. More precisely, we use the model called ‘MobileNetV3-Small’ [Howard et al., 2019] in its PyTorch [Paszke et al., 2019] implementation<sup>3</sup>. It requires rescaling the images from  $32 \times 32$  to  $224 \times 224$ , with a bi-linear interpolation, which is of course not optimal.

<sup>3</sup><https://pytorch.org/vision/main/models/mobilenetv3.html>

The network is used with *weights* (parameters of the network) obtained from an optimization (pre-training) based on a subset of 1,000 classes from the ImageNet data set [Krizhevsky et al., 2017]<sup>4</sup>. The dimension of the output vector is 1,000 (initial number of classes), additionally for some experiments we use a Principal Component Analysis (PCA) for dimension reduction. For the CIFAR-10 data set, the *training* set is composed of 50,000 samples and the *test* set of the remaining 10,000 samples. We compute the covariance matrix of the samples from the training set and then perform the dimension reduction through PCA on the training and test sets. We reduce the dimension from  $d = 1,000$  to  $d = 100$ . The 100 samples are *standardized i.e.*, subtracted by the mean and divided by the standard deviation of the training set.

The purpose of these experiments is not to obtain the best performance of classification but to show that the proposed implementation is able to produce reasonable predictions (in time and accuracy) compared to the tree-based methods, often used in the application to plankton images.

For all classifications methods, we used a CV on the training set to search for the best parameters. For the  $k$ -NN and  $W$ - $k$ -NN, we used  $CV_{LOO}$  with all possible  $k$  from 1 to  $k_{max} = 300$ . The best parameters are respectively  $k = 15$  and  $k = 20$ . For the Gradient boosting method (XGBoost implementation), we took advantage of the GPU support (used to estimate the best splits) and used a 5-folds CV. We tested 5 values of tree depth (3, 5, 7, 10, 15) and 3 learning rates (0.1, 0.5, 0.7) for 100 decision trees. The best couple of parameters was 7 for the depth and 0.5 for the learning rate. With the best parameters, we then computed the prediction (for all methods) on the test set. We observed a coherence with the CV scores (see table 3.2). As expected, the prediction time is shorter for the tree-based method by a factor  $\sim 3.5$ . The evolution of the ACC as a function of the number of neighbours  $k$  is given on appendix C.3 for the  $k$ -NN and  $W$ - $k$ -NN. As a conclusion, the  $k$ -NN and  $W$ - $k$ -NN implementations are competitive to the boosting method. It can be remarked that the  $CV_{LOO}$  estimations on neighbours-based methods are much faster than the 5-folds CV (XGBoost) with this implementation, which make them useful in practice.

Method	Prediction time [ms]	CV time [s]	CV methods	ACC [%]
XGBoost	35	458	5-folds	83.4 (83.5)
$k$ -NN	121	26.7	$CV_{LOO}$	82.1 (82.2)
$W$ - $k$ -NN	124	39.6	$CV_{LOO}$	82.7 (82.6)
$k$ -NN ( $d=1000$ )	861	29.6	$CV_{LOO}$	83.8 (84.6)
$W$ - $k$ -NN ( $d=1000$ )	886	44.5	$CV_{LOO}$	84.6 (84.3)

**Table 3.2** Summary of the results on the CIFAR-10 data set. The last column is the ACC score from the CV with the set of best parameters, in parentheses is the ACC on the test set.

For the sake of curiosity, we ran the same experiments for the  $k$ -NN and  $W$ - $k$ -NN on the *raw* extracted features of dimension  $d = 1000$ . The surprising result is that the predictions were more accurate in that high-dimensional space (see table 3.2 and figures in appendix C.3), while we were expecting the opposite. Indeed, the meaning of the Euclidean distance between sample in such a space is not trivial and the search of the neighbours was expected to face the so-called *curse of dimensionality*.

<sup>4</sup>More details in <https://github.com/pytorch/vision/tree/main/references/classification#mobilenetv3-large--small>.



## Chapter 4

# Experimental Results on Plankton Images

### **Key points – Main results on the classification of plankton images**

1. We present classification results with the proposed implementation for two real plankton data sets of reference.
2. We compare the results with a reference method used in practice (RF).

### **Contributions – Presentation of the data sets and description of the experiments**

3. Description of the image data sets
4. Classification with the  $W$ - $k$ -NN and a RF based on handcrafted features (ZooProcess), for both data sets.
5. Same classifications with image features extracted with a CNN.
6. Accurate binary classification of copepods *vs.* *others*.
7. Advantages and limits of the  $W$ - $k$ -NN in practice.

**Chapter 4 – Experimental Results on Plankton Images:**

4.1	Introduction . . . . .	42
4.2	ZooScan instrument & ZooScanNet data set . . . . .	42
4.2.1	Presentation . . . . .	42
4.2.2	ZooProcess: Handcrafted Features . . . . .	42
4.2.3	ZooScan image features from fine-tuned CNN . . . . .	46
4.3	UVP5-HD instrument & data set . . . . .	47
4.3.1	UVP5-HD <i>in situ</i> imaging instrument . . . . .	47
4.3.2	UVP5-HD image classification with features from a fine-tuned CNN . . . . .	48
4.3.3	UVP5-HD Copepods . . . . .	50

**4.1 Introduction**

In this chapter, we deal with two real-world plankton image data sets. The goal is to show that the proposed implementation of the  $W$ - $k$ -NN (chapter 3) is efficient with real, large, and complex classification tasks. For such classification tasks, the score to maximize is the Balanced Accuracy (B-ACC). It is the mean of the accuracies per class. Therefore, it is more adapted for our applications to plankton images, regarding the class imbalance (see figs. 4.2 and 4.5). The ACC will be also given as an indication. This holds for all the chapter.

**4.2 ZooScan instrument & ZooScanNet data set****4.2.1 Presentation**

The ZooScan instrument [Gorsky et al., 2010] (see fig. 4.1) is a widespread instrument ( $\sim 300$  units across the world) designed to produce images of zooplankton using a line scanner. Collected samples are placed on the scanner and all organisms are imaged together ( $\sim 1500$  per scan). Then, single image per organisms are isolated through segmentation. The instrument is produced by the company Hydroptic<sup>1</sup>. Its main specifications are given in table 4.1. While the main disadvantage of this instrument is to operate in the lab (as opposed to being *in situ*), it can produce high resolution images of zooplankton at a high rate. Its efficacy and widespread use resulted in large data sets. EcoTaxa contains over 35 million ZooScan images. It allowed taxonomists to manually label them and this resulted in the creation of a reference data set: ZooScanNet, containing 1.4 million labelled images, all checked by several experts [Elineau et al., 2018]. The finer taxonomic level is composed of 136 classes containing *living* and *not-living* objects, with an extremely unbalanced class distribution (see fig. 4.2; from a few samples per class to over a hundred thousands). In the following sections, we present some results of its classification with the  $W$ - $k$ -NN classifier. We use 70% of the data set to search for the best parameters of the classifiers via Cross-Validation (CV). The remaining 30 % are used as a test set to estimate the performances on *unseen* data.

**4.2.2 ZooProcess: Handcrafted Features**

In this section, we focus on the classification of the ZooScanNet images based on 46 handcrafted features extracted with the ZooProcess software (ImageJ plugin). The list of these image features and their associated definitions are given on fig. B.1 and table B.1).

<sup>1</sup>[http://www.hydroptic.com/index.php/public/Page/product\\_item/ZOOSCAN](http://www.hydroptic.com/index.php/public/Page/product_item/ZOOSCAN)

<sup>2</sup>From <https://lov.imev-mer.fr/web/facilities/piqv/>



Figure 4.1 ZooScan instrument, at LOV <sup>2</sup>.

Dimensions (LxWxH)	60 x 54 x 36 cm (Cover closed)
Weight	25 Kg
Input voltage	110 to 230 VAC, 50 to 60 Hz
Interface	USB 2.0
Robustness	Resistant to salt water, diluted formaldehyde and diluted ethanol (5%).
Specifications	Samples ZooSCAN is designed to handle and digitize liquid samples
Sample volume	0.2 litre to 1 litre
Specifications	Non-destructive with safe sample recovery High resolution, optimized for objects larger than 200 $\mu$ m in equivalent spherical diameter
Image resolution	up to 2200 dpi (dots per inch) Each image is 14,150 x 22,640 pixels and contains ~ 1500 individual animals. Each image is processed as a single frame of 24.5cm x 15.8cm. Optimized lighting system to enhance image quality and contrast supplied ZooScan is supported by a number of open-source computer programs that runs on Windows 10 : ImageJ with ZooProcess macros, EcoTaxa. ZooScan.

Table 4.1 ZooScan specifications, from Hydroptic

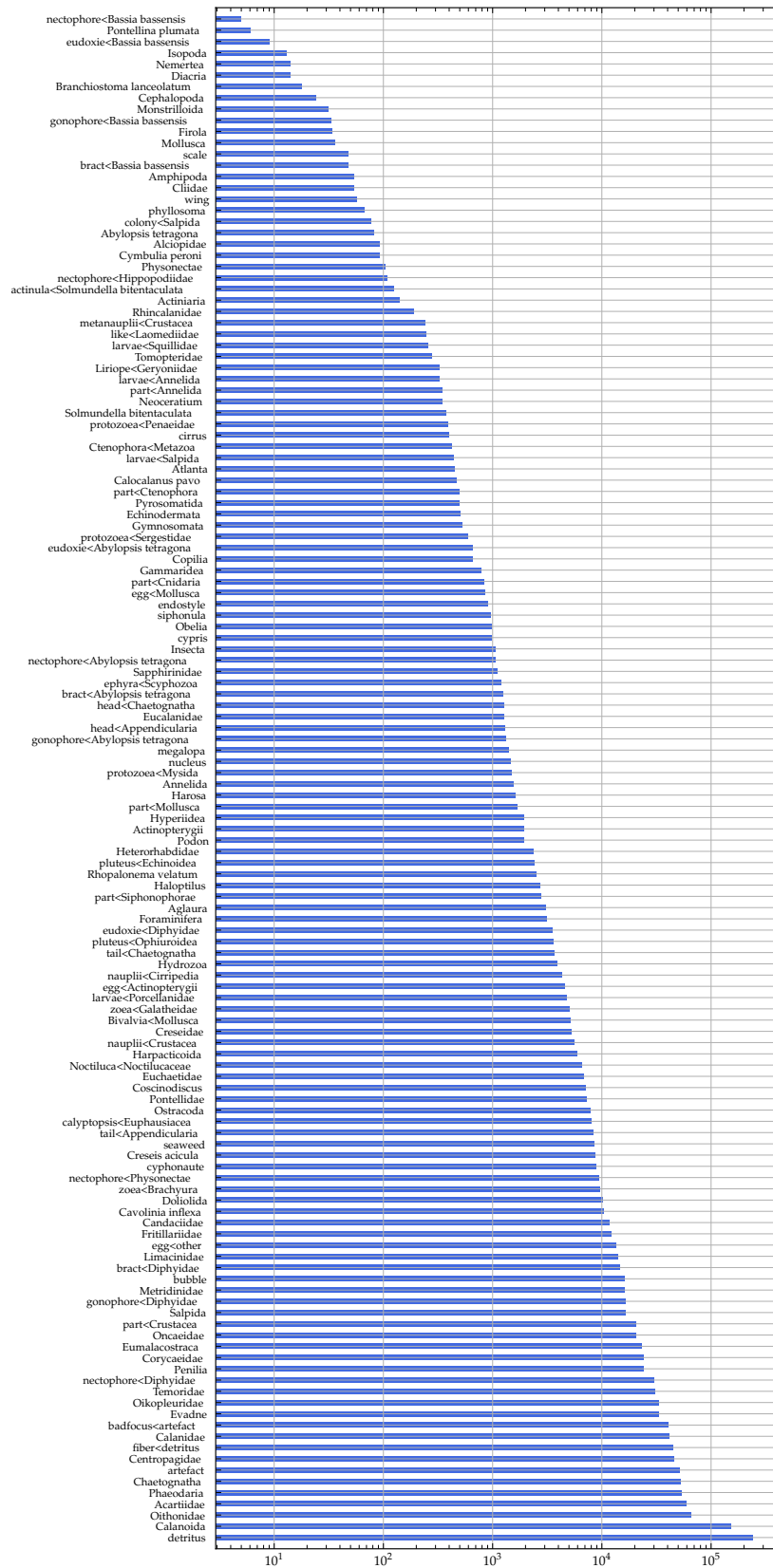
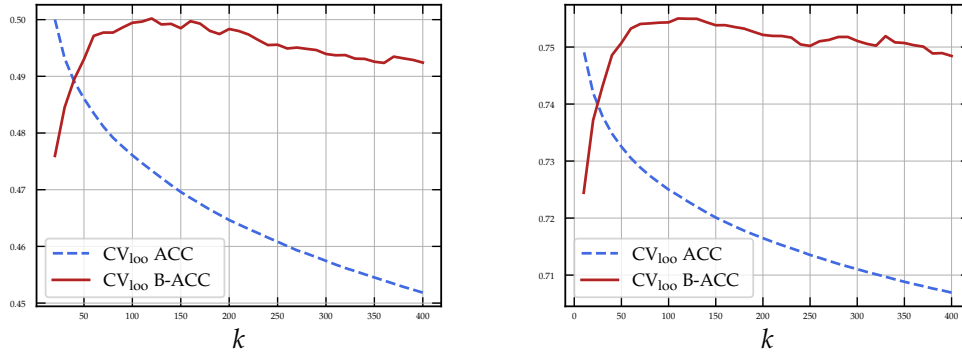


Figure 4.2 Ordered ZooScanNet classes counts. The count scale is logarithmic.

Using the implementation detailed in chapter 3, we computed the B-ACC and ACC scores for each sample using a  $CV_{LOO}$ , for 40 values of the parameter  $k$ , from 10 to 400 with a step of 10. The evolution of the scores with respect to the number of neighbours considered is given on fig. 4.3. The optimal parameter is the one that maximize the value of B-ACC and is  $k^* = 120$ .



**Figure 4.3** Evolution of the scores with respect to the number of neighbours  $k$  for the  $W$ - $k$ -NN. Left: based on features from ZooProcess;  $k$  from 10 to 400 with a step of 10; the optimal value is  $k^* = 120$ . Right: based on features from a fine-tuned CNN;  $k$  from 1 to 400, step of 10; the optimal value is  $k^* = 110$ .

The total computation took less than 40 minutes, keeping the sample weights in memory (see chapter 3). The prediction time on all the  $\sim 400,000$  test samples for  $k^* = 120$  was  $\sim 2$  minutes.

One main observation is that the scores are not very sensitive to the parameter  $k$  (at least in the computed range). Indeed, the maximum deviation between the scores is less than 5% (<3% for B-ACC). The decrease of the ACC score with the number of neighbours is certainly due to the over-representation of samples from the *detritus* class. For low value of  $k$  a strong weight is given to close neighbours, so it is likely to be classified as *detritus* since the probability to have *detritus* samples in the neighbourhood is high. Then, the ACC is ‘high’ because the *absolute* number of samples correctly classified is also high. Another way around is to note that if every sample was classified as *detritus*, the ACC would be high since the most of the samples (*i.e.*, *detritus*) would be correctly classified. This is why the accuracy normalized per class, *i.e.*, the B-ACC preferred for the taxonomic classification of plankton. Nevertheless, in practice, the *not-living* objects are the most abundant, so the user may want to have access to the ACC. Also, if the B-ACC score does not evolve (or almost not) with  $k$ , the ACC can help to pick the optimal value  $k^*$ . Those remarks also stand for the next experiments.

While the purpose here is to show the method is usable in practice, it is preferable to compare it (on the same machine) with a standard classifier used in practice by the users, typically at Laboratoire d’Océanographie de Villefranche (LOV): a Random Forest (RF) [Breiman, 2001]. It is important to note that, in practice, decision-tree-based methods may be tedious in the tuning of the parameters, mainly: impurity criterion, number of trees, maximum depth of the trees and the minimum number of samples required to compute a split in a decision tree. For this experiment, we limit the search of the optimal parameter of a RF to the latter and fix the formers. The search for the best split is done by minimizing the ‘gini’ criterion, with 100 trees. The maximum depth of the trees is only limited by the remaining parameter to tune: the minimum number of samples to perform a new split (*i.e.*, there is no limit in terms of absolute depth). We search for this best parameter among the values (10, 100, 1000) using a 3-folds CV. In addition, to compensate for the class imbalance, class weights are used. The ‘gini’ impurity (for each split in each tree) is computed with a weight per sample. This weight is inversely proportional to the number of



samples belonging to its class (used for this tree). The comparison, with the  $W$ - $k$ -NN, of the setting of the experiments is shown on table 4.2 (third column). The scores are given as an indication to show that both can produce similar results. The RF can probably achieve a better score by probing a larger parameter space (see a comprehensive experiment in Panaiotis et al. [in press]). This actually shows the practical issues of the methods with multiple parameters: *they are not trivial to optimize*. On the other hand, the proposed  $W$ - $k$ -NN relies on a unique, discrete parameter. Hence, it is simpler to set and to interpret (degree of confidence to the neighbourhood). About the fitting and prediction times, we can notice the prediction time of the RF method is shorter than for the  $W$ - $k$ -NN by a factor 4 to 8. This is due to the nearest-neighbour search, which is the main limitation of the proposed implementation. On the other hand, looking at the very large number of samples ( $\sim 400,000$  for the test set), the prediction time ( $< 2$  minutes) is relatively short for a laptop computer, thanks to the use of the GPU with the KeOps library. An advantage of the proposed implementation is its ability to compute the search of the optimal parameter using a  $CV_{LOO}$  in a relatively short time, compare to the RF that was only computed for 5 folds. This can be useful in practice, for example to test the influence of the features on the classification (adding or removing features), which ask to fit the model multiple times.

		ZooProcess, $d=46$	fine-tuned CNN + PCA, $d=10$
$W$ - $k$ -NN	tested parameters	40	40
	CV	$CV_{LOO}$	$CV_{LOO}$
	CV fitting time [h]	0.44	0.43
	pred. time [s] (test set)	119	57
	B-ACC [%] (test set)	49	75
RF	tested parameters	3	3
	CV	3-folds	3-folds
	CV fitting time [h]	0.35	0.29
	pred. time [s]	15	13
	B-ACC [%]	55	71

**Table 4.2** Summary of classification results for the ZooScan data set with the  $W$ - $k$ -NN and the RF methods.

### 4.2.3 ZooScan image features from fine-tuned CNN

To go a step further in the experimental setting, looking at what is done in practice at LOV, we propose to compute the classification of the same samples, based on image features extracted with a CNN. More precisely, the network ‘MobileNetV2’ [Sandler et al., 2018] is optimized on the ImageNet data set (see section 3.4) and ‘fine-tuned’ (*i.e.*, re-optimized with desired 136 classes), on a few hundreds of plankton images. From the so-called ‘deep-features’ we performed a dimension reduction with a PCA keeping only the 10 most relevant components. The PCA is based on the covariance matrix of the hundreds of samples used for the fine-tuning only, and applied to all the data.

We computed the same experiments as in the previous section. For the RF, the tested parameters (*i.e.*, minimum number of samples required to compute a split) were the same (10, 100, 1,000); 100 the best one. For the  $W$ - $k$ -NN, we tested all the values of  $k$  from 1 to 400 with a step of 10, see fig. 4.3 (right). The optimal parameter was  $k^* = 110$ . From table 4.2 the same conclusions can be drawn as in the previous section. It is interesting to note the gain in B-ACC (for both methods) with the features extracted with a fine-tuned CNN, with only ten dimensions. The confusion matrix and the recall score per class for the test set are given in figs. C.10 and C.11.

This show that our method can be used and is useful in practice on the ZooScan data set. In particular, it considerably simplifies the parameter tuning and can therefore help to explore different combinations of features more easily, in compare to other standards methods such as RF.

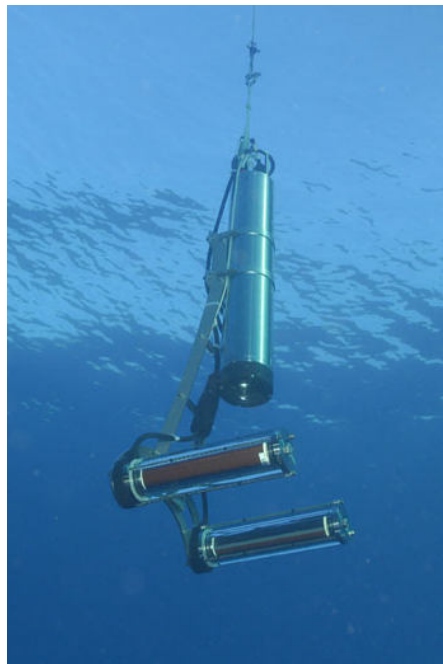
### 4.3 UVP5-HD instrument & data set

In this section, we present results similar to those of section 4.2 for a data set of 2-d *in situ* images taken with the instrument Underwater Vision Profiler 5 - High Definition (UVP5-HD).

#### 4.3.1 UVP5-HD *in situ* imaging instrument

Imaging instruments can be distinguished in two categories : *ex situ* , *i.e.*, samples are collected and analysed later in the lab (*e.g.*, ZooScan), and *in situ i.e.*, samples are directly imaged in their environment.

The UVP5 camera [Picheral et al., 2010] is an *in situ* instrument designed for imaging plankton, produced by Hydroptic<sup>3</sup> (see fig. 4.4 and table 4.3). It is able to light up a fixed volume of water (about 15cm × 20cm × 3.5cm which is ~ 1 litre) in order to image a set of focused objects. The camera is deployed from a ship with a winch, such that it image vertical profiles. These profiles are sets of images from the surface to a fixed depth (down to 6000 m, but generally <500 m). They are useful to characterize ecosystems.



**Figure 4.4** Photography of the UVP5-HD instrument; deployed in the bay of Villefranche-sur-mer. Credit David Luquet.

Multiple versions of this instrument exist. Here we focus the Underwater Vision Profiler 5 - High Definition (UVP5-HD). The UVP5-HD data set is composed of 3 million images organized

<sup>3</sup>[http://www.hydroptic.com/index.php/public/Page/product\\_item/UVP5\\_DISCONTINUED](http://www.hydroptic.com/index.php/public/Page/product_item/UVP5_DISCONTINUED)

Operational Depth	0 to 6000 meters
UVP Dimension (H)	110 cm
Weight in air	30 Kg
Input voltage	110 to 230 VAC, 50 to 60 Hz
Lighting	Red LED at 625 nm in two glass cylinders
Standard Image volume	1.02 litres per frame (about 15cm x 20cm x 3.5 cm)
Image resolution	Acquires images of objects > 100 $\mu$ m
Additional infos	Real time processing
Mount	Stand-alone, Rosette
	Capable of acquiring and processing images from the surface even in strong sunlight.
	UVP is supported by ZooProcess

**Table 4.3** UVP5-HD specifications, from Hydroptic

into 35 taxonomic classes. The number of classes is less than for the ZooScan (section 4.2.1) notably because the image definition of the instrument is lower. Hence, it is more difficult to guarantee the identification of the classes at a fine taxonomic level. In other words, there are all kinds of organisms (and *non-living* objects are still present) as for the ZooScan data set, but organized into fewer classes. The number of images per classes is even more imbalanced, see the distribution in fig. 4.5. As for the ZooScanNet, we take 70% of the data to search for the best parameters and the remaining 30% to test the performance.

### 4.3.2 UVP5-HD image classification with features from a fine-tuned CNN

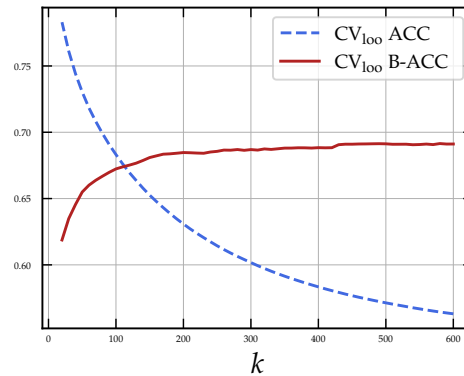
As we saw in the introduction (section 1.3.2), CNNs are able to extract coherent image features for the classification of plankton images. This is observed in section 4.2 and expected for the UVP5-HD data set. We followed the same pipeline of feature extraction as in section 4.2.3 (CNN fine-tuned on images of the training set and dimension reduction through PCA). The total time for all the  $CV_{LOO}$  estimations was about 2 hours, for  $k_{\min} = 10$ ,  $k_{\max} = 600$  with a step of 10. The parameter that maximized the B-ACC (69.2%) was  $k^* = 580$ , see fig. 4.6. On the test set, the score is 67.1%. Note the variation of the score is less than 10% for all the tested values of  $k$ , which is a hint on the stability of the method to its unique parameter. Further experiments with the investigation of larger values of  $k$  would be an improvement of the experiment, since we do not see a clear decrease of the B-ACC on fig. 4.6.

As a reference, we computed the classification with the RF method, on the same features. We used the same set-up as in section 4.2.3 except that, this time, the values for the minimum samples per split tested were 500, 1,000 and 5,000. The classification on the 3-folds gave a CV B-ACC score of 64.8% with the optimal parameter of 1,000 and 66.8% on the test set.

Figure 4.7 (another representation in fig. C.13) show the recall scores on the test set *i.e.*, the number of samples correctly classified in a class, relative to the number of elements in that class. The confusion matrix for the  $W$ - $k$ -NN predictions on the test set is given in fig. C.12. On fig. 4.7 we observe that both methods follow a similar pattern. The main difference is that RF seems to be more appropriated for the most represented classes (bottom of the figure), *e.g.*, copepods. On the other hand,  $W$ - $k$ -NN tends to classify more accurately the less represented classes. Drawing conclusions from this unique comparison would be hazardous. Indeed, it is not clear if this unique observation would generalize. Note that, for both methods, the weights used to re-balance the classification are inversely proportional to the number of samples per class. Nevertheless, they are not used in the same way. For the RF, it is used to search the best



**Figure 4.5** Ordered UVP5-HD classes count (~3 million objects). The count scale is logarithmic. *The not-living objects are distinguished with a \* (note their dominance).*



**Figure 4.6** Evolution of the scores with respect to the number of neighbours  $k$  for the  $W$ - $k$ -NN;  $k$  from 10 to 600, step of 10; the optimal value is  $k^* = 460$ .

split in a tree node, while for the  $W$ - $k$ -NN it is used to weight the contribution of each sample in the transformation (section 3.2.3).

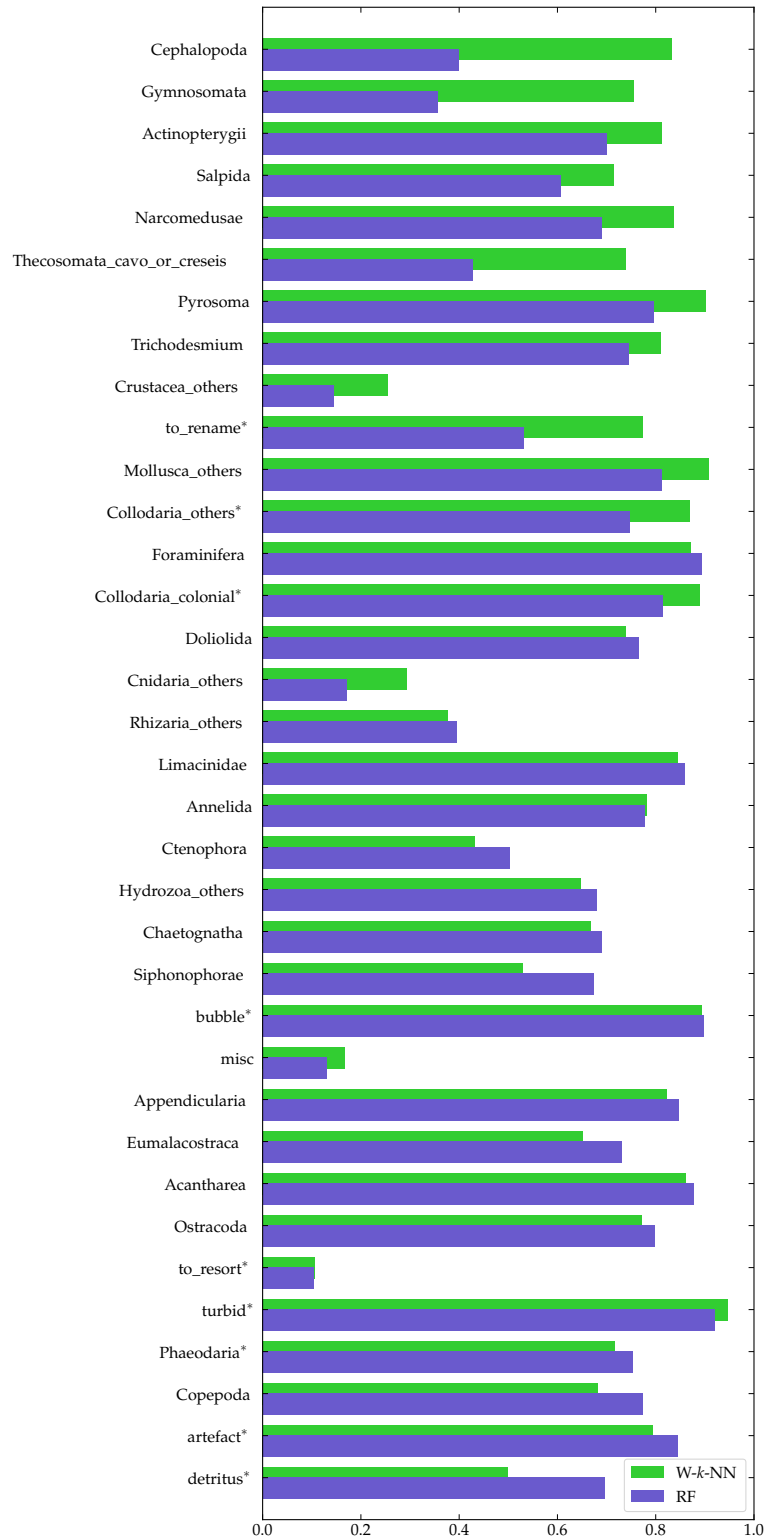
### 4.3.3 UVP5-HD Copepods

The identification of images of a specific group can be useful for ecological application. For example, the second part of this thesis relies on *in situ* 2-d images of copepods only. Here we focus on the identification of the copepods in the data set *i.e.*, binary classification copepods *vs. other*. We use the  $W$ - $k$ -NN to classify the images based on the images features extracted with the fine-tuned CNN. The hope is to get a higher recall for the copepod class in this binary setting (as opposed to the previous, multi-class setting).

The samples are the same as in the previous experiment (section 4.3.2). We work with the same sample-to-sample weights, that we stored in memory. This highlight the modularity of the proposed method. More precisely, based on the sample weights, different classification can be performed. This is an advantage of the method compared to other parametric methods (*e.g.*, SVM, RF). We computed the  $CV_{LOO}$  classification for  $k$  from 10 to 600 with a step of 10 (*i.e.*, same as before) in less than 4 minutes. The curve of the score is given in fig. 4.8. It looks like the B-ACC and the recall for the copepods continue to increase slightly for  $k > 600$ ; the optimal value was  $k = 600$ . Nevertheless, the B-ACC on the test set was 95% and the recall for the copepods was 96%. This is a high improvement compared to the recall in the multi-class setting (<70% for copepods, see fig. 4.7). This difference can be explained looking at the recall and precision scores<sup>4</sup> on fig. 4.9. Low values of precision and high values of recall translates a detection bias toward the copepods, *i.e.*, the classifier tends to over-predict samples as copepods. Indeed, high values of recall means the majority of the copepod samples of the data set are well identified as copepods; low values of precision means a large amount of non-copepod objects are predicted as copepods. This can be observed on figs. C.14 and C.15.

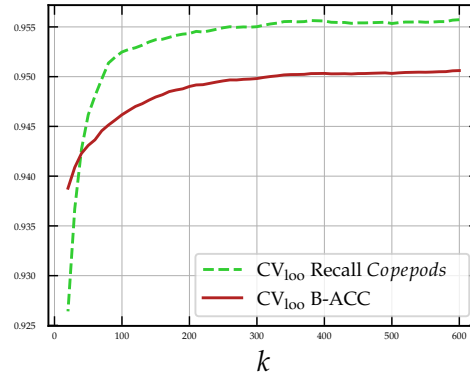
Similar classification performances are obtained with a RF classifier (same setting as in section 4.3.2): B-ACC of 95 % and recall on copepods of 94 % on the test set. Nevertheless, due to the optimization of the model, we could not rely on the previous RF model of the section 4.3.2

<sup>4</sup>Recall: number of samples correctly classified in a class, relative to the number of elements in that class.  
Precision: number of samples correctly classified in a class, relative to the number of elements predicted in that class.

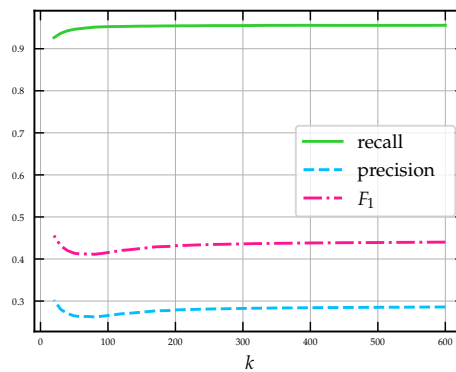


**Figure 4.7** Recall score on test set (number of samples correctly classified in a class, relative to the number of elements in that class) for the classification based on the first 10 components of the *deep*-features with the *W-k-NN* ( $k^* = 580$ ) and RF (minimum samples to compute a new split: 1000). The classes are ordered according to the number of samples, as in fig. 4.5 (less represented at the top). The classes that represent *not-living* objects are mentioned with the symbol \*.

to accelerate the procedure. Hence, it took 12 minutes to fit 3-folds with the 3 parameters (500, 1000, 5000).



**Figure 4.8** B-ACC and recall for the copepods class as a function of  $k$ .



**Figure 4.9** Recall, precision, and  $F_1$  scores for the copepod class as a function of  $k$ . The  $F_1$  score is the harmonic average of the recall and the precision scores.

The empirical results from this chapter demonstrate that (i)  $W$ - $k$ -NN can be used in practice on real-world plankton data sets of millions of images (*ex situ* and *in situ*), (ii) it is simple to tune with a unique parameter, (iii) it can produce accurate predictions, (iv) it is easy to understand since it relies on the similarity with the neighbours and (v) it is useful for classifying the same samples into various class groupings (*e.g.*, see section 4.3.3). On the other hand, its limitation is on the prediction. Indeed, it can be time-consuming, which is an important point for real-world applications.

For the plankton application, it would be benefic to enlarge the search for the optimal parameter  $k$  to larger values, given its evolution in figs. 4.3 and 4.6. In order to evaluate if the  $W$ - $k$ -NN can produce as accurate results as other standard methods, a comprehensive study would be necessary, notably including a full CNN classifier.

## Part II

# Correcting the Estimations of Copepod's Total Volume from 2-d *In Situ* Imaging





## Chapter 5

### Copepods' Bio-Volume Estimates from *In Situ* 2-d Images

#### **Key points – Copepods' volume estimations from 2-d images are biased**

1. We present the so-called 'biological carbon pump' and the role of plankton organisms. We introduce the use of plankton *in situ* imaging campaigns for estimating global scale biogeochemical processes. We motivate the need for accurate total volume estimations for the copepod group.
2. We highlight the limitations of volume estimation on 2-d images and bring forward the ellipsoidal model for copepods.

#### **Contributions – Highlight of the biases of two standard methods**

3. Description of the dataset used for total volume estimations of copepods.
4. Presentation of the two State-Of-The-Art methods for copepods volume measurement for 2-d images.
5. Demonstration of their limits with examples.
6. Computation of the total copepods' volume estimates and highlight of the discrepancy between them.

**Chapter 5 – Copepods' Bio-Volume Estimates from 2-d In Situ Images:**

5.1	Introduction . . . . .	56
5.2	Imaging the copepods worldwide: the UVP5-Cop dataset . . . . .	57
5.3	Standard geometrical measurements with ZooProcess . . . . .	57
5.4	Standard methods for volume estimations . . . . .	59
5.4.1	Optical model . . . . .	59
5.4.2	Using the equivalent spherical diameter ( $\mathcal{M}_{\text{ESD}}$ ) . . . . .	60
5.4.3	Using a best-fitting ellipse ( $\mathcal{M}_{\text{ELL}}$ ) . . . . .	60
5.5	Limits of the Standard Methods . . . . .	61
5.5.1	Illustration of the limits of the current method . . . . .	61
5.5.2	Illustration of $\mathcal{M}_{\text{ESD}}$ & $\mathcal{M}_{\text{ELL}}$ error . . . . .	61
5.5.3	Discrepancy between total volume estimations . . . . .	61

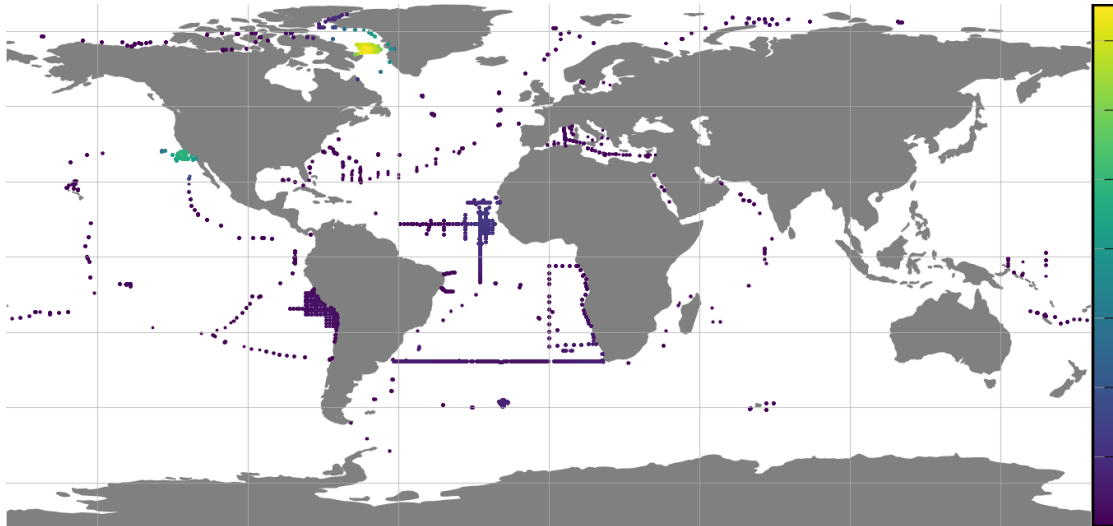
**5.1 Introduction**

Oceans cover 70% of the Earth surface. Global scale biogeochemical cycles, notably of carbon, are largely driven by the oceans. Let us remind that the phytoplankton organisms capture carbon through photosynthesis and zooplankton organisms aggregate and export it to the seabed (biological cycle). This process is referred as the 'biological carbon pump'. While these processes contribute to the regulation of climate, their quantification remains challenging, because of the scales it demands to deal with. Multiple methods have been developed to this end [Le Qu er  et al., 2015]. A key contribution was to highlight the high correlation of the zooplankton major trait, its size, with the carbon pump efficiency. More precisely, the relevant measure of size for these organisms is their volume, which is related to their biomass through their density. As their biomass increases, carbon sequestration increases too. In other words, zooplankton size can be seen as a 'proxy' for carbon sequestration in the ocean (biological contribution).

In this part of the thesis, we focus on the global estimation of zooplankton biomass. This was made possible by the world-wide *in situ* observation campaigns carried out those last decades [Kiko et al., 2022], opening the way for high-resolution density estimations per taxon, together with the taxonomic classification based on images, at the global scale. We deal with images of the UVP5 *in situ* camera (see section 4.3) that captures the *meso*-zooplankton (from  $\sim 0.1$  to  $\sim 1$  mm, mainly on the first 200 metres), where the carbon pump efficiency is at its maximum [Buitenhuis et al., 2006]. Among mesoplankton, the most numerous taxonomic group is *copepods*, which represents about 85% of the organisms for this layer (0-200 m) [Longhurst, 2007]. This can be observed on fig. 4.5, where the copepods constitute the most represented (living) group. For now, the sampling does not cover a large enough surface to conclude on the total number or volume of copepods with the raw data only. The map fig. 5.1 shows the repartition and density of copepods images (manually validated annotations) from the UVP5 at global scale. Nevertheless, combined with high-resolution environmental variables, inferences of the total volume per taxa are already accessible thanks to the comprehensive study by Drago et al. [2022] led by the LOV.

One main limitation of such works comes from estimation of individual organisms' volumes from 2-d images. Indeed, due to the projection onto the image plane, the true individual volume can not be computed. Instead, as we will see later in this chapter, estimations are made based on geometrical assumptions. Then, the total volume estimation is computed as the sum of the individuals (biased) ones. In this part of the thesis, we focus on the correction of the *total* volume estimations from 2-d *in situ* images, for the copepods. In this chapter, we will first present the dataset (copepods images from the UVP5 camera), followed by the State-Of-The-Art methods

for estimating the volume of a copepod from its projection onto the image plane. Then we will highlight the limits of those methods using a specific example, and finally, we will present the *raw* results of total copepod volume estimates, and their limits.



**Figure 5.1** World UVP5 copepod sampling. The colour gives a hint on the sampling density

## 5.2 Imaging the copepods worldwide: the UVP5-Cop dataset

Details about the UVP5 instrument can be found in section 4.3.

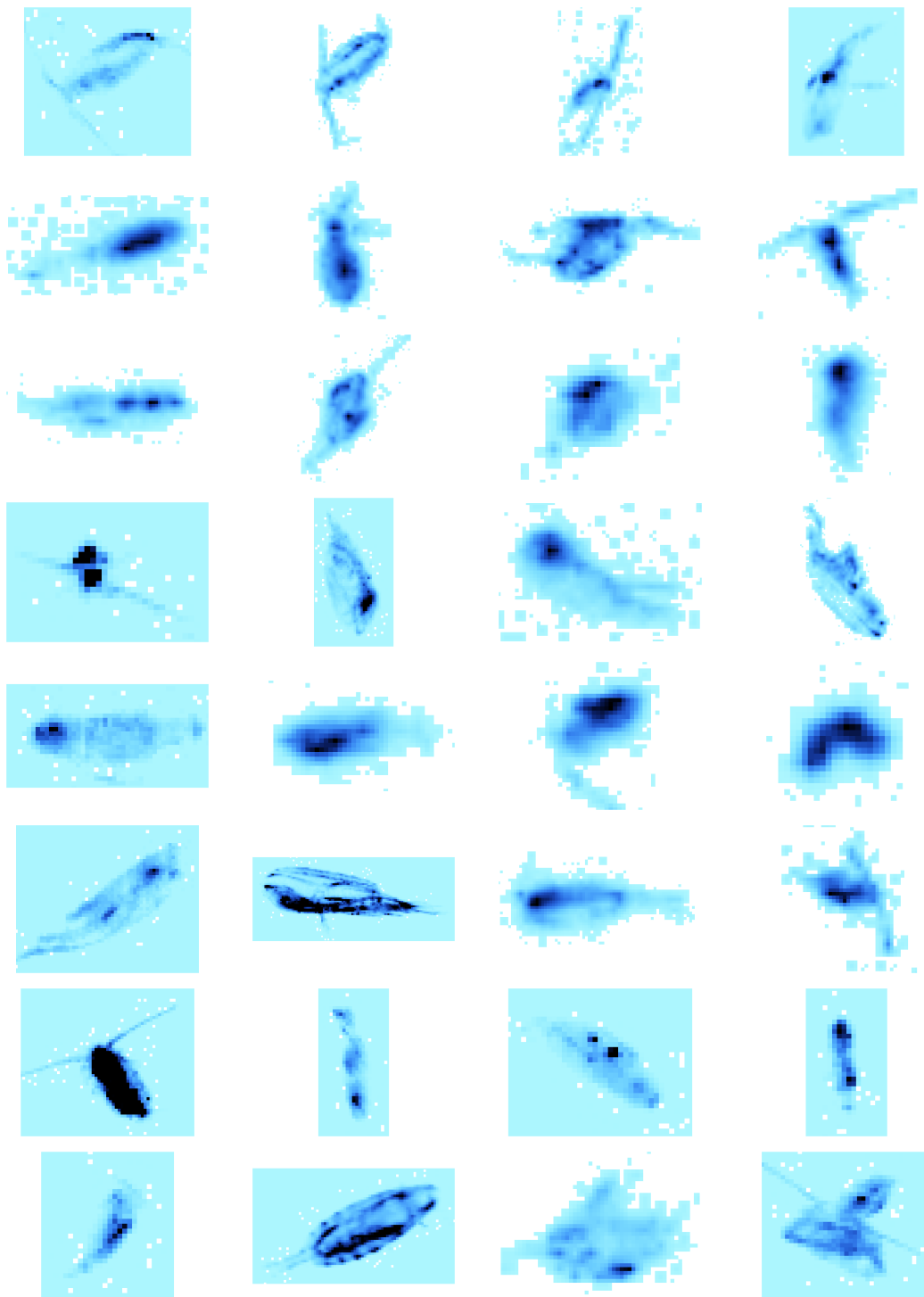
Following the UVP5 acquisition process, a segmentation is performed to extract small images containing a single organism (ideally, since organisms sometimes overlap). Metadata, such as geographical position, depth, and time of the acquisition are stored. All information are loaded on the EcoTaxa<sup>1</sup> web platform. Then, a taxonomic classification of the organisms is performed. First, a pre-trained model can be used to classify images and second, the label is validated or corrected manually by an expert.

The Underwater Vision Profiler 5 *Copepod* (UVP5-Cop) dataset is composed of all the validated images of copepods from the UVP5 instruments (SD & HD), that represent 158,487 samples. Images were processed following the new image processing method (see chapter 6). When the estimated volume (see section 5.4) was below  $0.1 \text{ mm}^3$ , the copepod silhouette detection was considered erroneous, and the image was excluded from the dataset. Images of partially cropped copepods (*i.e.*, with a part of the body outside the image) were also excluded, to avoid additional biases. Around 2,500 images were excluded (<2% of the data set), leaving 155,945 copepod images for analysis. Typical images of the dataset are shown on fig. 5.2. They are greyscale, with a pixel size varying from 0.086 to 0.174 mm depending on the generation and configuration of the UVP5. These pixel sizes are used to rescale all measurements to millimetres before processing.

## 5.3 Standard geometrical measurements with ZooProcess

To later infer the volume, we first need to isolate the copepod on the image. Then, we need to compute some geometrical quantities, such as the surface area of the copepod *prosome* (*i.e.*, its

<sup>1</sup><https://ecotaxa.obs-vlfr.fr/>



**Figure 5.2** Example of copepods images from UVP5. The colour represent the pixel intensity, it was chosen for clarity; the pixels of null intensity are white (background).

body) and its best-fitting ellipse. ZooProcess is a standard tool used by ecologists to achieve this [Gorsky et al., 2010; Picheral et al., 2010], based on ImageJ [Schneider et al., 2012]. The image is segmented through thresholding (the threshold may vary with the camera used) thus, yielding a binary mask.

**Area estimation** : The area of the copepod prosome is estimated as the number of pixels of the larger connected component of the binary mask, then converted to  $\text{mm}^2$  using the calibrated pixel size.

**Best-fitting ellipse** : The copepod prosome is estimated as the larger connected component of the same binary mask. The axes lengths of the best-fitting-ellipse are given by the eigenvalues ( $1/\sqrt{\lambda_i}$ ) of the covariance matrix of the pixel's position<sup>2</sup>. Then, they are scaled such that the area of the ellipse is the same as the binary mask. This is expected to reduce the volume estimation errors for wrong ellipses fits.

After both of these measurements, we will derive two estimations of the same underlying volume.

## 5.4 Standard methods for volume estimations

For estimating the volume from 2-d images, hypothesis on the third dimension are needed. The two standard volume estimation methods are presented below. They are based on the key observation that the shape of the (3-d) copepod's prosome is close to an ellipsoid of parameters  $r_1 \geq r_2 \geq r_3$  (see multiples point of view in fig. 5.2). Carefully note that, the projection of an ellipsoid onto a plane is an ellipse (this is show in chapter 7), such that the observed projection of the prosome can be modelled with an ellipse of parameters  $\rho_1 \geq \rho_2$ .

### 5.4.1 Optical model

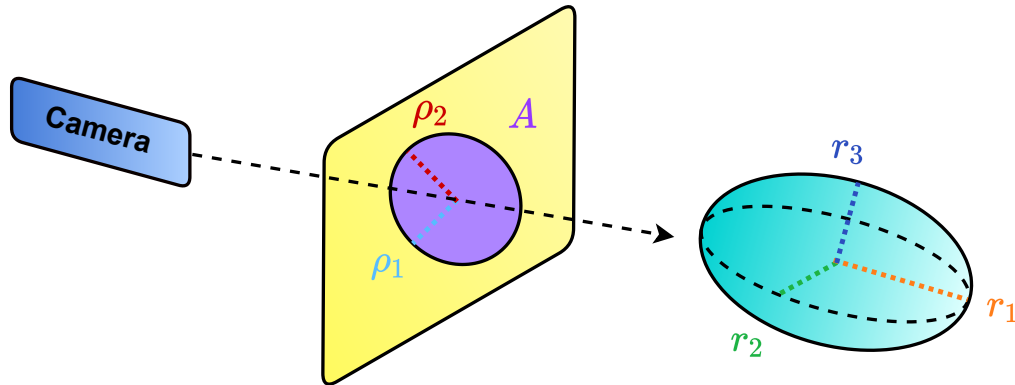
Let us start by defining the context of our applications. To safely infer a volume in three dimensions from a shape projected in two dimensions, the following assumptions are made:

- The distance between imaged objects and the camera is the same for all objects (or differences are negligible).
- One of the two following statements is true.
  - The size of the object is negligible compared to its distance from the camera. Hence, even if the camera has a perspective acquisition geometry, it can be approximated well enough by a parallel one.
  - The acquisition system follows a line scanner principle (then, its acquisition geometry is intrinsically parallel);

With these hypotheses, the imaging process can be schematically represented as in fig. 5.3 (with an ellipsoidal object). All in situ plankton imagers presented in [Lombard et al., 2019] (including ZooScan and UVP5) satisfy these conditions.

---

<sup>2</sup>The orientation is given by the eigenvectors, but is useless here.



**Figure 5.3** Representation of the geometrical setup of the imaging, with some notations:  $r_1, r_2,$  and  $r_3$  the true semi-axes of an ellipsoidal object, with  $r_1 \geq r_2 \geq r_3$  by convention;  $\rho_1, \rho_2$  the semi-axes of the projected ellipse, with  $\rho_1 \geq \rho_2$  by convention;  $A$  the area of the projected shape.

#### 5.4.2 Using the equivalent spherical diameter ( $\mathcal{M}_{\text{ESD}}$ )

There is a unique disk with the same area  $A$  as the organism's projected silhouette (the silhouette of the organism as observed on the image), and its diameter is  $\text{ESD} = 2\sqrt{\frac{A}{\pi}}$ . The Equivalent Spherical Diameter estimation method ( $\mathcal{M}_{\text{ESD}}$ ) makes the assumption that the volume of the organism can be approximated by the volume of the sphere of diameter ESD, that is

$$V_{\text{ESD}} = \frac{4}{3}\pi \left(\frac{\text{ESD}}{2}\right)^3. \quad (5.1)$$

If the organisms were indeed spherical and the 3-d -to- 2-d acquisition system performs a parallel projection, then  $V_{\text{ESD}}$  would be the exact volume.

For ellipsoidal objects, like copepods, the projection silhouette is an ellipse of semi-axes  $\rho_1$  and  $\rho_2$ , with  $\rho_1 \geq \rho_2$  by convention (see fig. 5.3). Its area is equal to  $\pi\rho_1\rho_2$ . Therefore, the equivalent diameter is  $\text{ESD} = 2\sqrt{\rho_1\rho_2}$ .

#### 5.4.3 Using a best-fitting ellipse ( $\mathcal{M}_{\text{ELL}}$ )

A common alternative to  $\mathcal{M}_{\text{ESD}}$  is to fit an ellipse shape on the projection and construct an ellipsoid in three dimensions ( $\mathcal{M}_{\text{ELL}}$ ). It should be more appropriate for objects of ellipsoidal shape, such as copepods (assuming the antennas and *urosoma* (*i.e.*, tail) are thin/small enough for their influence on the volume to be negligible). It proceeds as follows: (i) an ellipse is fitted on the object silhouette, defining two semi-axes:  $\rho_1$  and  $\rho_2$  (fig. 5.3), (ii) the smallest semi-axis of the fitted ellipse ( $\rho_2$ ) is duplicated to form the triplet of semi-axes of an ellipsoid, (iii) the volume is computed as

$$V_{\text{ELL}} = \frac{4}{3}\pi\rho_1\rho_2^2. \quad (5.2)$$

## 5.5 Limits of the Standard Methods

### 5.5.1 Illustration of the limits of the current method

The purpose of this section is to give an overview of the area estimations and ellipse fits implemented in ZooProcess. More details and argumentation will be given in chapter 6. Note that we took a threshold of 8 over 256 grey levels to produce the binary mask, and that other thresholds might be used by the user. On fig. 5.4, we give examples of randomly selected images from the UVP5-Cop dataset with their best-fitting ellipse and binary mask. This allows to study qualitatively the results (there are no quantitative results, since there is no ground truth is available). For example, on (a) the ellipse fit well to the copepod and the mask (b) gives a correct idea of the area of the copepod (at least what we can infer from the image). For the other examples((c) to (f)), the presence of antennas (and/or pixel noise), pollute the results (both ellipse fit and area estimation). Even if it is difficult to conclude on the quality of the results statistically, after looking at hundreds of random examples, it appears that the number of examples such as (c) and (e) is non-negligible. This motivates the implementation of a new method in chapter 6.

### 5.5.2 Illustration of $\mathcal{M}_{\text{ESD}}$ & $\mathcal{M}_{\text{ELL}}$ error

For this illustration, we model the copepod prosome in 3-d with an ellipsoid and assume the measured ESD and semi-axes of the ellipse are exact. With this set-up, the error of the  $\mathcal{M}_{\text{ESD}}$  estimation can range from large underestimation to even larger overestimation, depending on the orientation of the ellipsoid (fig. 5.5). Those individual errors transcribed to the total volume estimations, but the pending question is *'how does the total estimations are affected ?'*. The purpose of chapters 6 and 8 is to address this question.

For  $\mathcal{M}_{\text{ELL}}$ , despite the fact that the estimation of the silhouette shape is more appropriate than with  $\mathcal{M}_{\text{ESD}}$ , errors due to the projection from 3-d to 2-d are still present (fig. 5.5). We can remark that, within this ellipsoid model framework,  $V_{\text{ELL}}$  is always lower than or equal to  $V_{\text{ESD}}$ . Indeed, we imposed by convention that  $\rho_1 \geq \rho_2$ , therefore,

$$\begin{aligned} \sqrt{\rho_1 \rho_2} &\geq \rho_2 \\ \Leftrightarrow \sqrt{\rho_1 \rho_2}^3 &\geq \rho_1 \rho_2^2 \\ \Leftrightarrow V_{\text{ESD}} &\geq V_{\text{ELL}} \end{aligned}$$

with equality when the projection silhouette is a circle.

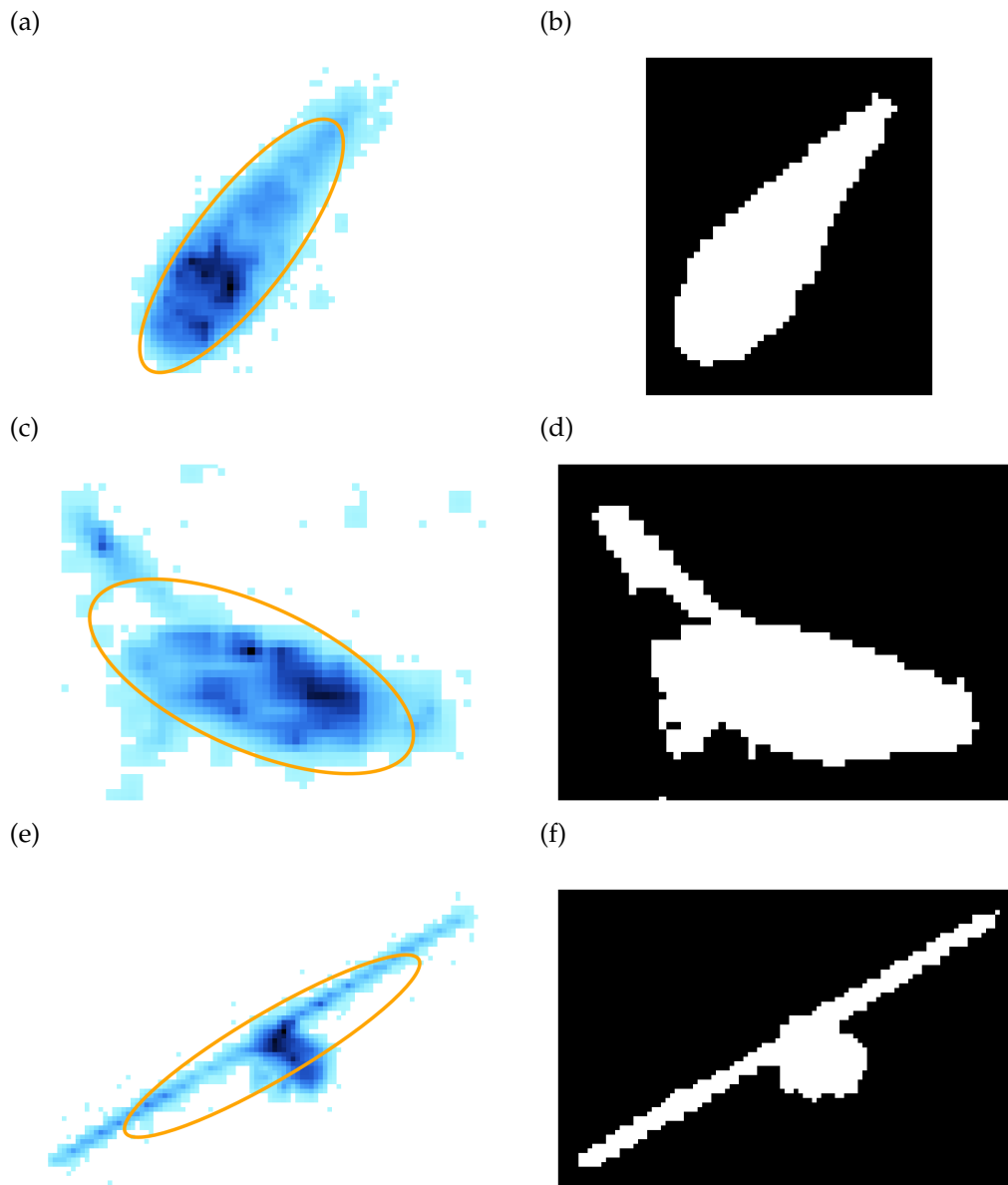
### 5.5.3 Discrepancy between total volume estimations

We saw two main error sources for the total copepod volume estimates. As a reference point, let us compute the total volume of copepods from the UVP5-Cop dataset with the State-Of-The-Art methods. The details of the methods are listed below, and the results are summarized in table 5.1.

- ZooProcess &  $\mathcal{M}_{\text{ESD}}$  : Area is extracted with ZooProcess, see section 5.3. The volume is computed according to eq. (5.1).
- ZooProcess &  $\mathcal{M}_{\text{ELL}}$  : The best-fitting ellipse is computed with ZooProcess, see section 5.3. The volume is computed according to eq. (5.2).

A useful observation is the gap between the results of table 5.1, which would be zero for exact volume estimation (or if both made the same error). In the following chapters, we will propose a

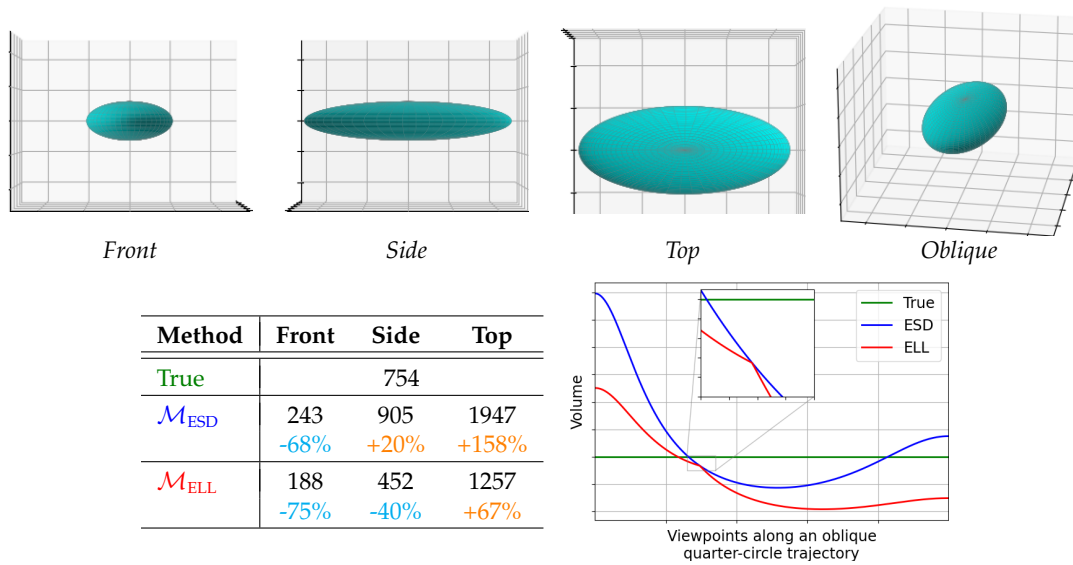




**Figure 5.4** Examples images of copepods, their ellipse fits obtained from ZooProcess (orange solid line), and binary mask. Note, even with the scaling of the ellipse area (section 5.5.3), the resulting ellipses do not fit the prosome because of the influence of the antennas.

$W_{\text{ESD}} [\times 10^5 \text{ mm}^3]$	$W_{\text{ELL}} [\times 10^5 \text{ mm}^3]$	Gap $[\times 10^5 \text{ mm}^3]$
7.51	4.79	2.72

**Table 5.1** Total volumes estimations  $W_{\text{ESD}}$  and  $W_{\text{ELL}}$  from  $\mathcal{M}_{\text{ESD}}$  and  $\mathcal{M}_{\text{ELL}}$  methods. The last column is the absolute difference between the estimations.



**Figure 5.5** Examples of volume estimations and errors made by  $\mathcal{M}_{\text{ESD}}$  and  $\mathcal{M}_{\text{ELL}}$  for an ellipsoid  $E$  with  $(r_1, r_2, r_3) = (1, 0.42, 0.25)$  (see fig. 5.3 for the definition of  $r_i$ ). The errors are computed from the analytic expressions of the projected semi-axes  $(\rho_1, \rho_2)$  detailed in chapter 7. The first row displays the simulated ellipse from various viewing angles. The table gives, in black, the rounded values of the volume for each method and, in colour, the percentage of under/over estimation compared to the true value, which is computed from the  $r_i$ s and  $\rho_i$ s. The lower right plot shows the volume computed with each method for viewpoints regularly sampled along an arc turning around the ellipsoid; note that the  $\mathcal{M}_{\text{ESD}}$  estimation is always greater than or equal to the  $\mathcal{M}_{\text{ELL}}$  one.

new method for the area extraction and the ellipse fit of the copepod's prosome (in chapter 6), and a statistical correction for the total volume estimation from  $\mathcal{M}_{\text{ESD}}$  and  $\mathcal{M}_{\text{ELL}}$  (in chapters 7 and 8).



## Chapter 6

# Extracting the Copepod Prosome from 2-d Images

### **Key points**

1. We propose a new procedure for fitting an ellipse and estimating the area of the copepod's prosome only.
2. We apply it to a real dataset and compare the results with the State-Of-The-Art .

### **Contributions**

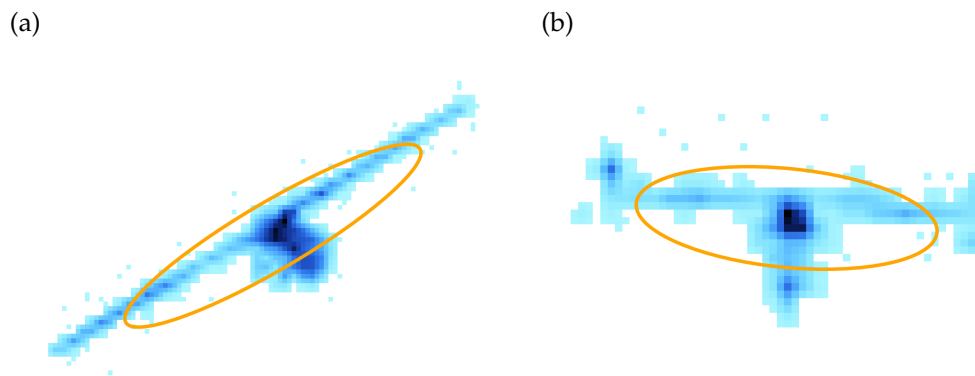
3. Method for extracting the copepod's prosome from 2-d images.
4. Computation of total copepods' volume estimates.
5. Illustration of the benefits.

**Chapter 6 – Extracting the Copepod Prosome from 2-d Images:**

6.1	Motivations : influence of antennas . . . . .	66
6.2	Geometrical measurements . . . . .	66
6.2.1	Common process . . . . .	66
6.2.2	Area estimation and ellipse fitting . . . . .	67
6.3	Application to UVP5-Cop images . . . . .	68
6.4	Discussion . . . . .	68

**6.1 Motivations : influence of antennas**

Clearly, copepod antennas, when visible, can affect, sometimes dramatically, the measurement of the projected area of their prosome (which constitutes the bulk of their volume) and/or the fitting of an ellipse to their silhouette (see fig. 6.1). Therefore, in the current chapter, we propose area estimation and ellipse fitting approaches tailored to copepods, to mitigate that phenomenon. The area estimation is based on a procedure that mimics mathematical morphology opening that first performs an erosion to discard the antennas and then a dilation to recover the area of the copepod body. Ellipse fitting is also performed after this opening-like operation.

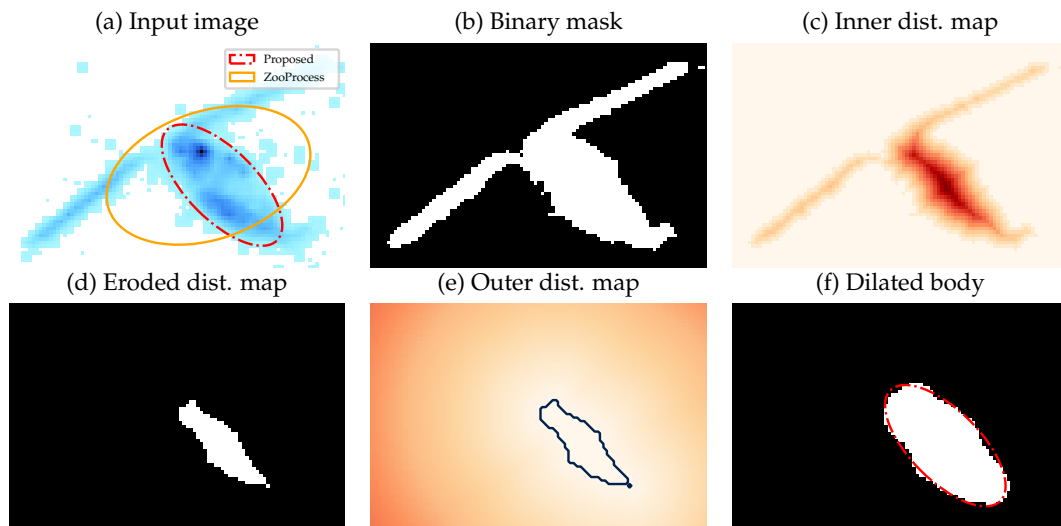


**Figure 6.1** Examples images of copepods and their ellipse fits obtained from Zooprocess (orange solid line). Note, even with the scaling of the ellipse area (section 5.5.3), the resulting ellipses do not fit the prosome because of the influence of the antennas.

**6.2 Geometrical measurements****6.2.1 Common process**

The general idea of the improved methods proposed in this chapter is to get rid of the antennas (and *urosome*, *i.e.*, the tail) before measuring the copepod silhouette surface or fitting an ellipse onto it. It is assumed that the binary mask of the copepod has been determined previously. We propose to compute the Inner Distance Map (IDM) of this mask and to erode it using a threshold. We fix the threshold to  $\max(\text{IDM}) \times 0.5328$  in our experiments, based on the visual results. This step allows getting rid of the antennas. Note that the binary mask could have been eroded directly using mathematical morphology. However, it would make use of a discrete so-called structuring element (typically a discretized disk), which would lead to a coarser eroded shape, given the small size of a copepod in our images (and hence, the small size of the discretized

structuring element). This could have a negative impact on the subsequent steps. Next, to recover the original copepod body size, the outer distance map of the eroded mask is computed and thresholded using the same threshold as the one used for erosion. This amounts to dilate the eroded mask, but again in a finer way than if using mathematical morphology. The various steps are illustrated in fig. 6.2. An implementation with *python* is available at the Inria GitLab<sup>1</sup>.



**Figure 6.2** Copepod body mask computation as a common preliminary step for surface estimation and ellipse fit. Reading the figure in lexicographical order, each image is the result of the processing of the previous one. They are: (a) the input greyscale image, (b) the binary mask obtained by thresholding, (c) the inner distance map, (d) the eroded mask obtained by thresholding, (e) the outer distance map, and (f) the dilated mask obtained by thresholding (same threshold). The orange ellipse is fitted on image (b) and rescale to the area of the mask (b), while the red ellipse is fitted on the proposed body mask (f).

### 6.2.2 Area estimation and ellipse fitting

The copepod surface estimation is performed by counting the number of pixels of its binary mask. The improved version simply counts the pixels of the mask obtained previously in 6.2 as opposed to counting the pixels in the original binary mask which includes the antennas.

When an object is described by a binary mask of pixels, the most classical ellipse fitting method interprets the pixels as the samples of a point cloud. The covariance matrix of the cloud is computed. Its eigenvectors represent the best fitting ellipse orientation, while its eigenvalues represent the semi-axes of the ellipse. A simple improvement of this method (or any other ellipse fitting method, as a matter of fact) consists in rescaling the fitted ellipse so that its area matches the object area. This is implemented by the software ImageJ that ZooProcess uses. However, if this improvement allows correcting the fitted ellipse surface (which can be enough for some applications), it does not help that much for copepod volume estimation. Indeed, the precision of the small semi-axis is crucial, and it is not improved by the surface adjustment. As a reminder,

<sup>1</sup><https://gitlab.inria.fr/cedubois/Copepod-Volume-Correction>

the  $\mathcal{M}_{\text{ELL}}$  estimation of the volume is:

$$V_{\text{ELL}} = \frac{4}{3}\pi\rho_1\rho_2^2 = \frac{4}{3}\underbrace{\pi\rho_1\rho_2}_{\text{Surface}} \underbrace{\rho_2}_{\text{Minor semi-axis}}. \quad (6.1)$$

Whatever the ellipse fitting method is, the starting point is the copepod mask. The standard fitting method get distracted by the antennas, which can result in very bad ellipses (see the orange ellipses in figs. 6.2 to 6.4). Therefore, we proposed to fit an ellipse on the mask obtained in section 6.2 instead of the original binary mask (see the red ellipses in figs. 6.2 to 6.4).

### 6.3 Application to UVP5-Cop images

Some carefully chosen examples of ellipse fits are shown on fig. 6.3 and others randomly selected on fig. 6.4. When antennas are visible, the proposed method fits better the prosome compare to the Zooprocess one. The limitation of the proposed method is illustrated on fig. 6.3 panel (b), where the best-fitted ellipse seems smaller than the actual copepod body. This is due to the threshold used for the erosion (and dilatation) of the IDM, that was previously fixed. The total volume estimations using the proposed image processing step for both methods are given on table 6.1, using the same dataset as for the previous estimations of total volume in section 5.5.3. While no ground truth is available, it can be observed that the gap between both estimations ( $\mathcal{M}_{\text{ESD}}$  and  $\mathcal{M}_{\text{ELL}}$ ) reduces by a factor 1.68 using our image processing method compared to the Zooprocess one. Knowing our process is motivated by the morphology of the copepods and looking at the qualitative results on randomly selected examples (fig. 6.4), we believe the new ellipses fit better the prosome of the copepod compared to the original method and result in a more accurate estimation of the total volume of copepods.

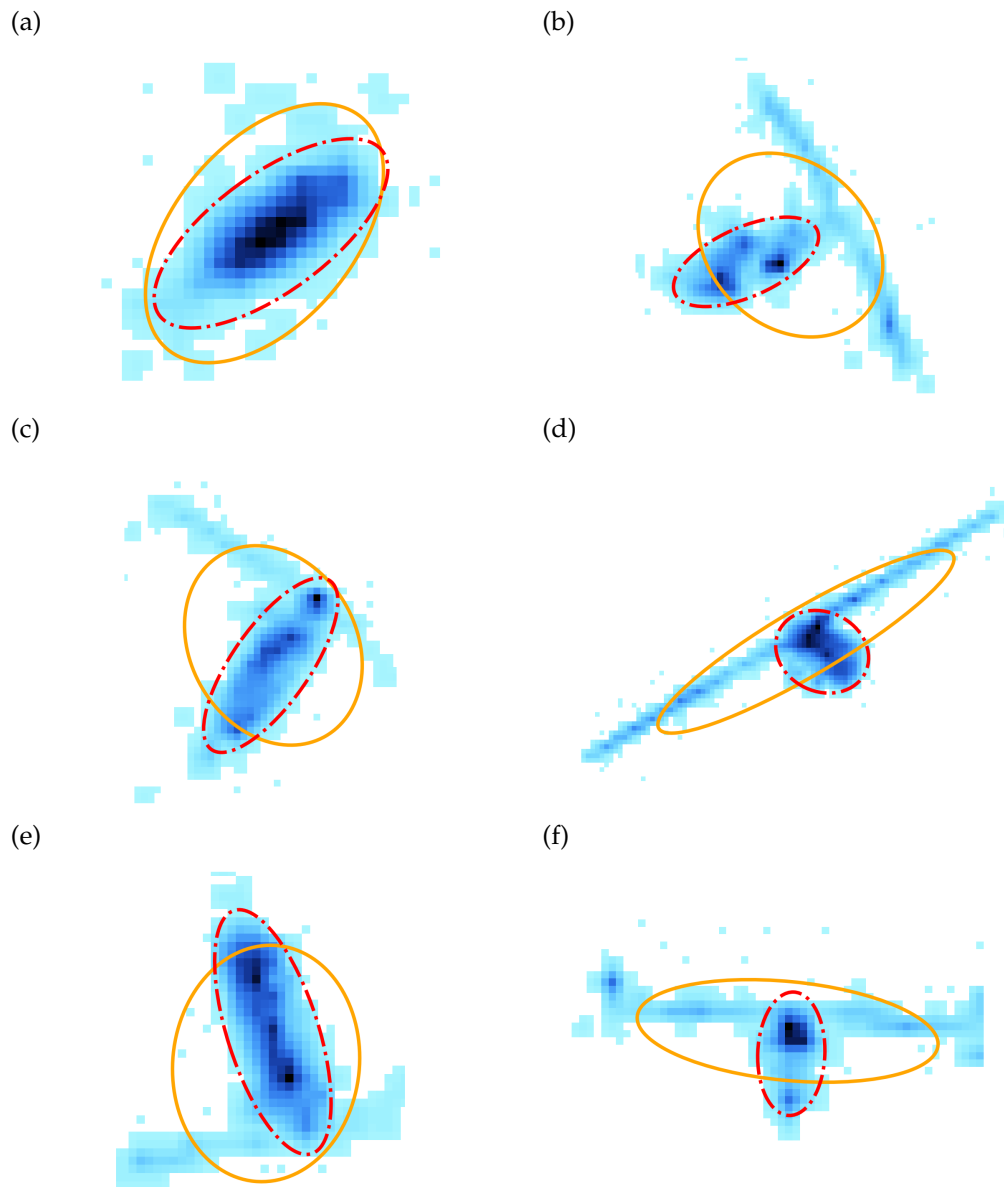
Method	$W_{\text{ESD}} [\times 10^5 \text{mm}^3]$	$W_{\text{ELL}} [\times 10^5 \text{mm}^3]$	Gap $[\times 10^5 \text{mm}^3]$
Zooprocess	7.51	4.79	2.72
Proposed fit	5.57	3.95	1.62

**Table 6.1** Total volumes estimations  $W_{\text{ESD}}$  and  $W_{\text{ELL}}$  from  $\mathcal{M}_{\text{ESD}}$  and  $\mathcal{M}_{\text{ELL}}$  methods. The first row is the same as table 5.1 *i.e.*, computed with Zooprocess. The second row corresponds to the proposed method. The last column is the absolute difference between the estimations.

### 6.4 Discussion

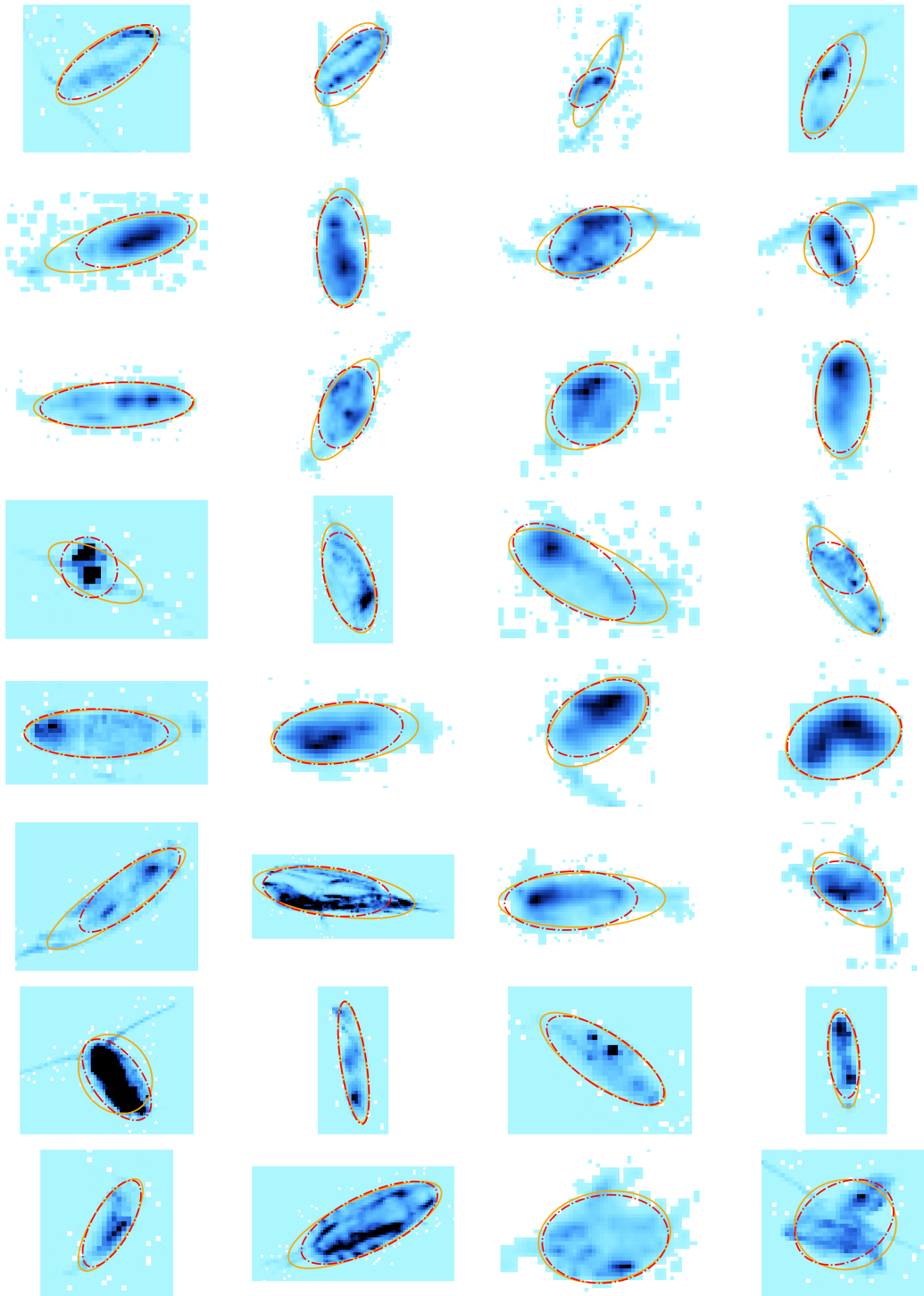
In this chapter, we proposed a method to estimate the projected area and to fit an ellipse on the copepod body only, to avoid large biases caused by the antennas (and, more rarely, the urosome). Neglecting the volume of the antennas and urosome compared to that of the prosome seems appropriate in first order, and it is essentially what the standard ellipse fit does when it is not affected by antennas. Still, the validity of this assumption would need to be tested. The volumes of these different parts seem difficult to measure experimentally but could be assessed from detailed 3D scans of individuals, which we now have the technology for.

An alternative to the standard and proposed methods, could be to fit the ellipse on the greyscale version of the object, that is, using the pixel intensities as sample weights when computing the covariance matrix. However, we found that this alternative does not work well on the copepod images of our dataset.



**Figure 6.3** Examples images of copepods and their ellipse fits. On these projections, the bodies of copepods can be well approximated by ellipses, which we assume to be ellipsoids in 3 dimensions (see appendix D). Ellipse fits on copepod images are obtained by Zooprocess [Gorsky et al., 2010] (orange solid line), affected by the antennas, and our method (dashed-dotted red), that fits the prosome of the copepod better.





**Figure 6.4** Multiple randomly selected examples of ellipse fit based on the original mask (Zooprocess); orange solid line and for the proposed method; dashed-dotted red. We see that when antennae are not visible, the result is almost the same, but when they are visible, the classic method is not appropriate.

Another aspect to test is whether our method indeed fits the projected silhouette better than the standard one. We noticed it does on many images similar to figs. 6.3 and 6.4. A small proportion of the results in fig. 6.4 still present wrong ellipse fits, which illustrates that there is a place for improvement. An expected one is to adjust the threshold of the erosion (and dilatation) automatically, depending on the size of the object or the image, for example. The fact that we observed a significant reduction of the discrepancy between the total volume estimated with  $\mathcal{M}_{\text{ESD}}$  and  $\mathcal{M}_{\text{ELL}}$  when using this approach compared to the classic one (see table 6.1) also suggests a gain in accuracy. However, to assess its absolute performance, a ground-truth segmentation should be performed on numerous images, by having human operators delineate the prosome of the copepod. Then, a pixel-level match between this ground-truth and the two automated approaches (Zooprocess and ours) could be computed. This extremely labor-intensive effort is considered to be out of the scope of this thesis.



## Chapter 7

# Modelling the projection of a copepod's prosome

### Key points

1. The prosome (body) of copepods is modelled with an ellipsoid. We derive its projection onto a plane, which is an ellipse.
2. Individual error of the volume estimation is scale invariant.
3. First characterization of the error.

### Contributions

4. Derivation of the exact projection of an ellipsoid
5. Proof of the invariance to scaling of the individual volume estimation error
6. Measures of real shape parameters
7. Simulation of realistic ellipsoids
8. Distribution of the error

**Chapter 7 – Modelling the projection of a copepod's prosome:**

7.1	Introduction . . . . .	74
7.2	Projection of an ellipsoid . . . . .	74
7.2.1	Geometrical setup . . . . .	74
7.2.2	Ellipsoid silhouette in 3-d . . . . .	75
7.2.3	2-d silhouette . . . . .	76
7.2.4	Semi-axes for parallel projection . . . . .	77
7.3	Volume estimation errors . . . . .	77
7.3.1	Expressions for the standard methods . . . . .	77
7.3.2	Invariance of errors to scaling . . . . .	78
7.4	Simulation of realistic ellipsoids . . . . .	79
7.4.1	A simulation for the total volume correction factors . . . . .	79
7.4.2	Estimating shape parameters . . . . .	81
7.4.3	Distribution of individual errors . . . . .	81

**7.1 Introduction**

The purpose of this chapter and the following (chapter 8) is to compute correction factors for the total volume estimates from the  $\mathcal{M}_{\text{ESD}}$  and  $\mathcal{M}_{\text{ELL}}$  methods, which are both biased by the projection. A first step is to derive the effect of the projection of the copepod onto the image plane. To this end, we propose to model the copepod's prosome (*i.e.*, body) by an ellipsoid. We derive its exact projection onto a plane, depending on its shape and orientation. This allows to compute the exact error of the individual volume estimation from both  $\mathcal{M}_{\text{ESD}}$  and  $\mathcal{M}_{\text{ELL}}$  methods, for given parameters (shape and orientation). For the total volume error (*i.e.*, the subject of interest), we present a pipeline to simulate realistic ellipsoids (based on manual measures) in various orientations, to compute their projection, and to estimate the correction factors.

**7.2 Projection of an ellipsoid****7.2.1 Geometrical setup**

A centred ellipsoid is defined by all 3-d vectors  $x$  verifying

$$x^T M x = 1, \quad (7.1)$$

where  $M$  is a positive definite<sup>1</sup>  $3 \times 3$ -matrix whose elements are denoted by  $m_{ij}$ . More details are given in appendix D. Let  $(i, j, k)$  denote an orthonormal basis and let  $O$  denote the origin (see fig. 7.1). To study how this ellipsoid projects onto a plane using perspective projection, let us define (*i*) an optical centre  $e$

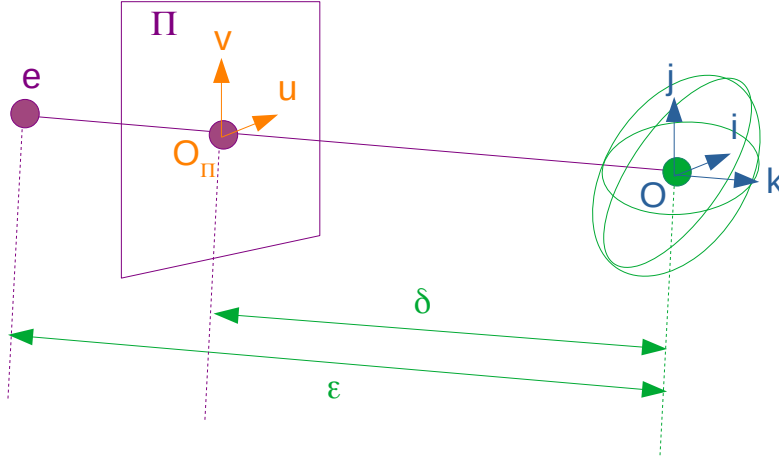
$$e = \begin{bmatrix} 0 \\ 0 \\ -\epsilon \end{bmatrix} \quad (7.2)$$

where  $\epsilon > 0$  is such that  $e$  is outside the ellipsoid, and (*ii*) a projection plane  $\Pi$  described by its normal

$$n = \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} \quad (7.3)$$

<sup>1</sup>positive definite matrices are, by definition, symmetric

and its distance to the origin  $\delta$ ,  $\epsilon > \delta > 0$ , such that  $\Pi$  does not intersect the ellipsoid. The plane  $\Pi$  is equipped with the orthonormal basis  $(u, v)$  where  $u$  and  $v$  correspond to  $i$  and  $j$  respectively. Its origin  $O_\Pi$  is located at the intersection between  $\Pi$  and the segment linking  $e$  to  $O$ . All these elements are illustrated on fig. 7.1.



**Figure 7.1** Geometrical setup of the ellipsoid model and its projection onto a plane. The camera is represented by its optical centre  $e$ , the sensor plane  $\Pi$ , and the sensor orthonormal coordinate system  $(O_\Pi, u, v)$ . The global orthonormal coordinates system is represented by  $(O, i, j, k)$ . The axes  $i$  and  $u$  are parallel, and so are  $j$  and  $v$ . The ellipsoid centre is at distance  $\delta$  from  $\Pi$  and  $\epsilon$  from  $e$ . Without loss of generality (for our problem), the ellipsoid centre is at  $O$ , and  $e$  is on the axis  $k$  with the optical axis aligned with  $k$ . Consequently,  $O_\Pi$  is also on the axis  $k$ .

### 7.2.2 Ellipsoid silhouette in 3-d

For some unit vector  $d$ , let  $x$  be defined as

$$x = e + \tau d \quad (7.4)$$

with  $\tau > 0$  and  $d \cdot n > 0$ . The ellipsoid silhouette as seen from  $e$  is given by the set of vectors  $d$  such that the half-line described by  $x$  when  $\tau$  varies is tangent to the ellipsoid<sup>2</sup>.

The point  $x$  is on the ellipsoid if and only if

$$(d^\top M d)\tau^2 + (2d^\top M e)\tau + (e^\top M e - 1) = 0, \quad (7.5)$$

which is of the form

$$\alpha\tau^2 + \beta\tau + \gamma = 0. \quad (7.6)$$

Therefore, the half-line described by  $x$  is tangent to the ellipsoid if and only if eq. (7.6) has a unique solution<sup>3</sup>, that is if and only if  $\beta^2 - 4\alpha\gamma = 0$ , which is equivalent to

$$d^\top S d = 0 \quad (7.7)$$

where  $S$  is equal to

$$S = M e e^\top M + (1 - e^\top M e)M. \quad (7.8)$$

<sup>2</sup>for such a vector  $d$ , there is a corresponding value for  $\tau$

<sup>3</sup>for completeness, note that this unique solution is  $\tau = -\frac{\beta}{2\alpha}$ .

The ellipsoid silhouette is defined by the solutions of eq. (7.7) respecting, as mentioned earlier, the following two conditions:  $|d| = 1$  and  $d \cdot n > 0$ .

Let  $S_\epsilon$  be defined as

$$S_\epsilon = \frac{1}{\epsilon^2} S. \quad (7.9)$$

Note that  $S_\epsilon$  can be used in eq. (7.7) in place of  $S$ .

In appendix D.2, the following (block matrix) expression is derived,

$$S_\epsilon = \left[ \begin{array}{c|c} M_{21}^\top M_{21} - m'_{33} M_{11} & (1/\epsilon^2) M_{21}^\top \\ \hline (1/\epsilon^2) M_{21} & (1/\epsilon^2) m_{33} \end{array} \right], \quad (7.10)$$

with  $m'_{33} = m_{33} - \frac{1}{\epsilon^2}$ .

### 7.2.3 2-d silhouette

As mentioned earlier, eq. (7.7) defines the silhouette of the ellipsoid in a perspective projection setup. On plane  $\Pi$ , this silhouette is defined by points  $p$  such that, for all solutions  $d$  to eq. (7.7),

$$\begin{cases} p = e + \tau' d \\ (p - e) \cdot n = \epsilon - \delta \end{cases} \quad (7.11)$$

where  $\tau'$  is a scalar<sup>4</sup>. The first equation of (7.11) is equivalent to

$$d = \frac{1}{\tau'} (p - e). \quad (7.12)$$

Equation (7.7) can now be rewritten in terms of  $p$  (and  $S_\epsilon$  as noted earlier) as follows

$$(p - e)^\top S_\epsilon (p - e) = 0. \quad (7.13)$$

The point  $p$  can be written as

$$p = O_\Pi + q \quad (7.14)$$

where  $q$  is a vector whose third component is equal to 0. Using a block formulation, we have

$$q = \begin{bmatrix} q_1 \\ 0 \end{bmatrix} \quad (7.15)$$

and

$$S_\epsilon = \left[ \begin{array}{c|c} S_{11} & S_{21}^\top \\ \hline S_{21} & s_{33} \end{array} \right]. \quad (7.16)$$

Then, eq. (7.13) is equivalent to

$$q_1^\top P q_1 + Q q_1 = r \quad (7.17)$$

where<sup>5</sup>

$$P = S_{11}, \quad (7.18)$$

$$Q = 2(\epsilon - \delta) S_{21}, \quad (7.19)$$

$$r = -(\epsilon - \delta)^2 s_{33}. \quad (7.20)$$

One recognizes the equation of an ellipse which can be put into the following standard form

$$(q_1 - c)^\top \left( \frac{1}{r - Qc/2} P \right) (q_1 - c) = 1 \quad (7.21)$$

where  $c = -P^{-1}Q^\top/2$  is the centre of the ellipse.

<sup>4</sup>combining the two equations of (7.11) together, one gets  $\tau' = (\epsilon - \delta)/(d \cdot n)$

<sup>5</sup>in case  $P$  is definite negative,  $P$ ,  $Q$  and  $r$  must be replaced with their opposite

### 7.2.4 Semi-axes for parallel projection

In appendix D.3, the expressions for the semi-axes  $\rho_i$  for perspective projection are given from the eigenvalues of  $P$ . For a parallel projection (*i.e.*,  $\epsilon = \infty$ ), the semi-minor and semi-major axes have the following simpler expression

$$\rho_i = \sqrt{\frac{m_{33}}{\lambda_i}}, i \in \{1, 2\} \quad (7.22)$$

and

$$P = m_{33}M_{11} - M_{21}^T M_{21}, \quad (7.23)$$

$$\lambda_i = (\text{tr}(P) + \sigma_i \sqrt{\Delta})/2, \quad (7.24)$$

$$|\sigma_i| = 1 \text{ and } \sigma_1 \sigma_2 = -1, \quad (7.25)$$

$$\Delta = \text{tr}(P)^2 - 4 \det(P) \quad (7.26)$$

where  $\text{tr}(P)$  is the trace of  $P$ ,  $\det(P)$  is its determinant, and the  $\sigma_i$ 's are chosen so that  $\rho_1 \geq \rho_2$ .

From these developments, we therefore have analytical definitions of an ellipsoid, its volume, its projection as an ellipse, the semi-axes and area of this projected ellipse.

## 7.3 Volume estimation errors

With the expression of the semi-axes, the analytical expression of the errors from the  $\mathcal{M}_{\text{ESD}}$  and  $\mathcal{M}_{\text{ELL}}$  estimation methods can be easily computed.

### 7.3.1 Expressions for the standard methods

For an axis-aligned ellipsoid,  $M$  is diagonal. The diagonal components are related to the semi-axes  $r_i$  as follows

$$m_{ii} = r_i^{-2}, i \in \{1, 2, 3\}. \quad (7.27)$$

The ellipsoid volume is then classically given by

$$V = \frac{4}{3} \pi r_1 r_2 r_3 \quad (7.28)$$

$$= \frac{4}{3} \frac{\pi}{\sqrt{m_{11} m_{22} m_{33}}}. \quad (7.29)$$

For a general ellipsoid (*i.e.*, any orientation), the volume is

$$V = \frac{4}{3} \frac{\pi}{\sqrt{\det(M)}} \quad (7.30)$$

where  $\det(M)$  is the determinant of  $M$ . Let  $V_*$  denote an estimation of the true volume  $V$ , where  $*$  is ESD or ELL here. The relative error in volume estimation is defined as

$$\mathcal{E}_* = \frac{V_*}{V}. \quad (7.31)$$

To write eq. (7.31) for  $\mathcal{M}_{\text{ESD}}$ , it should be reminded that, since the projection silhouette is an ellipse of area  $\pi \rho_1 \rho_2$ , the equivalent radius is equal to  $\sqrt{\rho_1 \rho_2}$ . Then, the relative errors of the



$\mathcal{M}_{\text{ESD}}$  or  $\mathcal{M}_{\text{ELL}}$  methods are

$$\mathcal{E}_{\text{ESD}} = (\rho_1 \rho_2)^{3/2} \sqrt{\det(M)} \quad \text{See eq. (5.1)} \quad (7.32)$$

$$\mathcal{E}_{\text{ELL}} = \rho_1 \rho_2^2 \sqrt{\det(M)}. \quad \text{See eq. (5.2)} \quad (7.33)$$

The following section demonstrates the invariance of these individual errors to scaling, a result that strongly impacts the practical procedure to simulate ellipsoids (presented in section 7.4).

### 7.3.2 Invariance of errors to scaling

#### Common remarks

The purpose is to show that  $\mathcal{E}_{\text{ESD}}$  and  $\mathcal{E}_{\text{ELL}}$  do not depend on the absolute volume of the ellipsoid, which is a function of  $(r_1, r_2, r_3)$ , but rather on the ellipsoid's proportions  $(r_2/r_1, r_3/r_1)$ . One way to prove this statement is to show that  $\mathcal{E}_*(\alpha M) = \mathcal{E}_*(M)$  for any  $\alpha > 0$ . Indeed, if this holds, then choosing  $\alpha$  equal to  $r_1^2$  implies that  $\alpha M$  is defined by the triplet  $(1, r_2/r_1, r_3/r_1)$ .

Let  $\rho$ , resp.  $\lambda$ , be a generic notation for  $\rho_1$  and  $\rho_2$ , resp.  $\lambda_1$  and  $\lambda_2$ . The other useful reminders are

$$\rho = \sqrt{\frac{m_{33}}{\lambda}} \quad (7.34)$$

$$\lambda : \text{eigenvalue of } P \quad (7.35)$$

$$P = m_{33}M_{11} - M_{21}^T M_{21}. \quad (7.36)$$

Let us add a subscript  $\alpha$  to these quantities to denote their expressions when  $M$  is replaced with  $\alpha M$ . We have

$$m_{33,\alpha} = \alpha m_{33} \quad (7.37)$$

$$P_\alpha = \alpha^2 P. \quad (7.38)$$

It is also clear that if  $\lambda$  is an eigenvalue of  $P$ , then  $\beta\lambda$  is an eigenvalue of  $\beta P$  ( $Px = \lambda x \Rightarrow \beta Px = \beta\lambda x$ ) for any  $\beta \neq 0$ . Therefore,

$$\lambda_\alpha = \alpha^2 \lambda. \quad (7.39)$$

Hence, it can be concluded that

$$\rho_\alpha = \frac{\rho}{\sqrt{\alpha}}. \quad (7.40)$$

Finally, note that we have the following property on the matrix determinant

$$\det(\alpha M) = \alpha^3 \det(M) \quad (7.41)$$

if  $M$  is a  $3 \times 3$ -matrix.

#### $\mathcal{M}_{\text{ESD}}$ method

As a reminder, the relative error in volume estimation of the  $\mathcal{M}_{\text{ESD}}$  method is

$$\mathcal{E}_{\text{ESD}}(M) = \mathcal{E}_{\text{ESD}}(r_1, r_2, r_3, \theta) \quad (7.42)$$

$$= (\rho_1 \rho_2)^{3/2} \sqrt{\det(M)}, \quad (7.43)$$

where  $\theta$  encode the orientation of the ellipsoid. Then

$$\mathcal{E}_{\text{ESD}}(\alpha M) = (\rho_{1,\alpha}\rho_{2,\alpha})^{3/2}\sqrt{\det(\alpha M)} \quad (7.44)$$

$$= \frac{(\rho_1\rho_2)^{3/2}}{\sqrt{\alpha^3}}\sqrt{\alpha^3\det(M)} \quad (7.45)$$

$$= \mathcal{E}_{\text{ESD}}(M). \quad (7.46)$$

### $\mathcal{M}_{\text{ELL}}$ method

As a reminder, the relative error in volume estimation of the  $\mathcal{M}_{\text{ELL}}$  method is

$$\mathcal{E}_{\text{ELL}}(M) = \mathcal{E}_{\text{ELL}}(r_1, r_2, r_3, \theta) \quad (7.47)$$

$$= \rho_1\rho_2^2\sqrt{\det(M)}. \quad (7.48)$$

Then

$$\mathcal{E}_{\text{ELL}}(\alpha M) = \rho_{1,\alpha}\rho_{2,\alpha}^2\sqrt{\det(\alpha M)} \quad (7.49)$$

$$= \frac{\rho_1\rho_2^2}{\alpha\sqrt{\alpha}}\sqrt{\alpha^3\det(M)} \quad (7.50)$$

$$= \mathcal{E}_{\text{ELL}}(M). \quad (7.51)$$

This concludes on the scaling invariance of errors in individual volume estimation. Thus, with the appropriate size normalization  $\alpha = 1/r_1$ , the error  $\mathcal{E}_*$  can be computed in terms of  $r_2/r_1$  and  $r_3/r_1$  only (which will be useful for the total volume correction in chapter 8).

## 7.4 Simulation of realistic ellipsoids

### 7.4.1 A simulation for the total volume correction factors

The theoretical total volume error is

$$\mathcal{T}_* = \frac{\iiint_{E_r} \int_{E_\theta} p(r, \theta) V_*(r, \theta) \, dr d\theta}{\iiint_{E_r} p(r) V(r) \, dr} \quad (7.52)$$

where  $E_r$  is the domain of ellipsoid semi-axes  $r = (r_1, r_2, r_3)$  ( $\{(r_1, r_2, r_3) \in \mathbb{R}^3 | r_1 \geq r_2 \geq r_3 > 0\}$ ),  $E_\theta$  is the domain of ellipsoid orientations,  $p(r, \theta)$  is the probability of observing a 'copepod ellipsoid' with size  $r$  and orientation  $\theta$ , and  $p(r)$  is the size probability. For both methods, it is

$$\mathcal{T}_{\text{ELL}} = \frac{\iiint_{E_r} \int_{E_\theta} p(r, \theta) \rho_1(r, \theta) \rho_2^2(r, \theta) \, dr d\theta}{\iiint_{E_r} p(r) V(r) \, dr}, \quad (7.53)$$

$$\mathcal{T}_{\text{ESD}} = \frac{\iiint_{E_r} \int_{E_\theta} p(r, \theta) \sqrt{\rho_1(r, \theta) \rho_2(r, \theta)^3} \, dr d\theta}{\iiint_{E_r} p(r) V(r) \, dr}. \quad (7.54)$$

Given the complexity of the terms involved in eqs. (7.53) and (7.54), no attempt was made to derive an analytical expression. Instead, in this section, we propose to simulate numerous random ellipsoids that realistically represent the body shapes of copepods, as well as their exact projection, to finally infer the correction factors (*i.e.*, to calculate numerically the integrals). Then they will be applied in the chapter 8.

The simplest choice to generate ellipsoids would be to draw the semi-axes from uniform distributions within appropriate ranges<sup>6</sup> and the orientation uniformly. However, those parameters should ideally be adapted to the data set at hand, since copepods' sizes, shapes, and acquisition viewpoints (or orientations) are related to environmental and imaging conditions. In the following, we propose a way to generate ellipsoids that follow a realistic, parametric copepod body model.

To generate random ellipsoids, one can directly generate random matrices  $M$ . Alternatively, random axis-aligned ellipsoids can be generated, which are then randomly rotated. The advantage of this later procedure is that constraints are more easily imposed to create copepod bodies with realistic proportions.

Axis-aligned ellipsoids are defined by the three semi-axes  $r_1$ ,  $r_2$ , and  $r_3$  (see details in appendix D). Since ellipsoids will be randomly rotated next, these values can be chosen such that  $r_1 \geq r_2 \geq r_3$  without loss of generality. To generate random values of  $r_1$ ,  $r_2$ , and  $r_3$ , we need to define one Probability Density Function (PDF, or 'distribution') per  $r_i$ . These PDFs must be defined in accordance with the reality of copepods' body shapes, either generically (*e.g.*, according to the literature) or from the data at hand. Generating an ellipsoid then amounts to drawing one random semi-axis value per PDF. If the semi-axes respect the ordering condition, then the ellipsoid is validated; otherwise, it is discarded, and a new round of random drawing must be performed. As such, the random process is of dimension three ( $r_1, r_2, r_3$ ) with two conditions ( $r_1 \geq r_2$  and  $r_2 \geq r_3$ ).

Fortunately, the process can be simplified by noting that the error made by the  $\mathcal{M}_{\text{ESD}}$  or the  $\mathcal{M}_{\text{ELL}}$  method on the total volume estimation is (almost) invariant to size normalization  $1/r_1$ . This is assumed for this chapter and show empirically in section 8.1.2. In other words, the error computed from  $N$  ellipsoids defined by  $(r_{1,n}, r_{2,n}, r_{3,n}), n \in [1..N]$  is (almost) equal to the error computed with the same ellipsoids, each scaled by the constant  $1/r_{1,n}$ . This amounts to normalizing the ellipsoids so that their largest semi-axis is equal to one (*i.e.*, defined by  $(1, r_{2,n}/r_{1,n}, r_{3,n}/r_{1,n})$ ). Thus, the random process becomes two-dimensional (defined only by the axes ratios  $r_2/r_1$  and  $r_3/r_1$ ) with only one condition ( $r_2/r_1 \geq r_3/r_1$ ); the three per-axis PDFs are replaced by two axes-ratio PDFs. This has two nice consequences: a statistical one and a practical one. Statistically speaking, to describe a random process through simulation, one needs 'exponentially' more samples as the dimension increases (this is known as the curse of dimensionality). The number of samples,  $N$ , is limited by computational constraints; thus, reducing the dimension provides a higher quality description for the same  $N$ . Practically speaking, the shift from drawing semi-axes to drawing semi-axes *ratios* means that the proposed method only depends on the *shape* of copepods (prosome height over prosome length and prosome width over prosome length), not on their overall size, which can be considered more general (size varies across regions) and more stable.

The remaining question is '*how to define the PDFs of the semi-axes ratios  $r_2/r_1$  and  $r_3/r_1$ ?*'. As for the per-axis PDFs, two reasonable options are literature-based and data-based. The literature may provide enough details to choose a PDF family (*e.g.*, Gaussian) and set the parameters for each ratio (*e.g.*, mean and variance for Gaussian). Alternatively, the ratios can be measured on physical samples or on images in which the copepods are seen from the side ( $r_2/r_1$  ratio) and from the top or bottom ( $r_3/r_1$  ratio). Then, the required PDFs can be fitted on these measurements in a parametric (*e.g.*, Gaussian, Beta, Gaussian mixture [Redner and Walker, 1984]) or non-parametric way (*e.g.*, Kernel Density Estimation (KDE) [Parzen, 1962; Scott, 1979]).

Finally, these ellipsoids, generated to match the body shapes of copepods, must be rotated to simulate a random acquisition viewpoint. In the absence of a strong *a priori* on the orientation

---

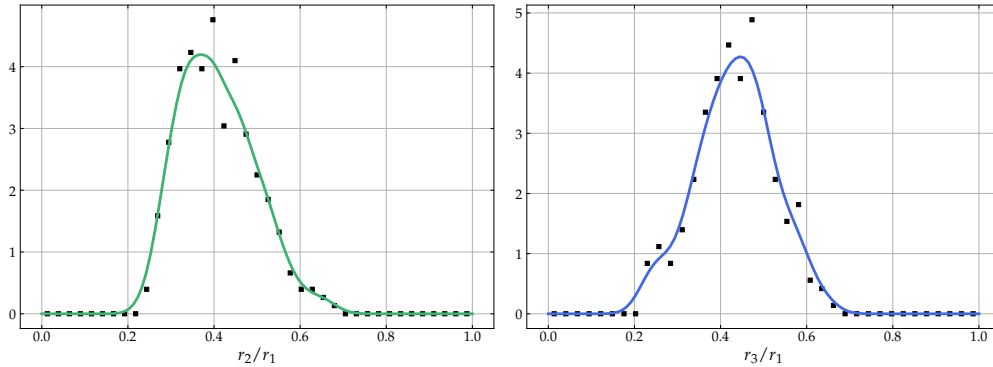
<sup>6</sup>valid ranges can be found in the literature *e.g.*, Conway [2012]

of copepods relative to the camera, these rotations can simply be uniformly random, in all directions. But the procedure can easily be adapted to generate rotations favouring preferred orientations (see section 8.3).

#### 7.4.2 Estimating shape parameters

As explained in the previous section, simulating copepod bodies requires PDFs for the two semi-axes ratios  $r_2/r_1$  and  $r_3/r_1$  and the rotation angle. Here we focus on the application with the UVP5-Cop dataset. We assumed uniform random orientation of copepods in the water column (see computational details in appendix E.3), since the data set is large, covers different depths and locations on the globe, and we do not know of any justification for a preferred orientation relative to the camera.

A set of axes ratios were measured on 295 images in which copepods were seen from the side and 265 images in which they were seen from the top or bottom. To gather these samples, operators manually selected images in which the orientation of the copepod was clear. The selection was guided to obtain a distribution in latitude similar to that of the whole data set, and distributions in length similar between the side and top/bottom samples (see appendix F). The constraint on latitude was meant to avoid biasing the samples towards a particular environment, since the morphology of copepod's varies latitudinally. The constraint in prosome length should ideally have been checked against the whole data set, to avoid estimating the axes ratios on biased samples within the  $\sim 150k$  simulated images. However, the true prosome length, or  $2 \times r_1$  in the 3-d ellipsoid, is unknown; only  $2 \times \rho_1$ , the major axis of the projected ellipse, can be estimated. While  $r_1 \simeq \rho_1$  in the side and top/bottom views,  $r_1 > \rho_1$  in any other view, so only the distribution of  $\rho_1$  in the side and top/bottom views can be compared. The PDFs for  $r_2/r_1$  and  $r_3/r_1$  were then estimated from the measurements using a KDE with a Gaussian kernel of optimal width [Scott, 1979] (see fig. 7.2).

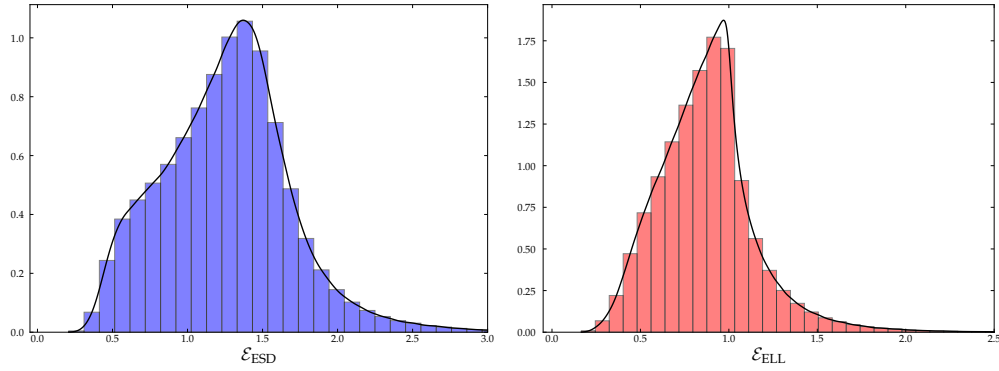


**Figure 7.2** Distributions of  $r_2/r_1$  (a) and  $r_3/r_1$  (b) fitted on our dataset. The markers are the normalized histograms of the samples, and the solid lines are the Gaussian Kernel Density Estimates.

#### 7.4.3 Distribution of individual errors

From a simulation of  $N=10^6$  ellipsoids with axes ratios distributed according to fig. 7.2, we computed the distributions of the error, shown on fig. 7.3. It seems that the distribution of  $\mathcal{E}_{\text{ESD}}$  is less centred on one (no error) than the one of  $\mathcal{E}_{\text{ELL}}$ . This let us conclude that more credit should be given to the  $\mathcal{M}_{\text{ELL}}$  method compare to the  $\mathcal{M}_{\text{ESD}}$  one, at least for the individual estimations. Also, the total volume estimated from  $\mathcal{M}_{\text{ESD}}$  is expected to be overestimated (looking at fig. 7.3),

while it is unclear to conclude for  $\mathcal{M}_{\text{ELL}}$  ones. The following chapter provides a clearer picture of those observations.



**Figure 7.3** Normalized histogram of the individual error  $\mathcal{E}_{\text{ESD}}$  ( $\mathcal{E}_{\text{ELL}}$ ) in blue (red) and associated KDE (black solid lines), obtained from the simulation of  $N=10^6$  ellipsoids ( $r_1 = 1$ ). The extrema of the error are 0.21 (0.17) and 5.31 (4.11).

## Chapter 8

# Correcting Total Copepods' Volume Estimates

### Key points

1. The purpose of this chapter is to compute and apply the correction factors for the total volume estimations of copepods in the UVP5-Cop data set.
2. We discuss the limitations and the robustness of the method.

### Contributions

3. Empirical study of the invariance of the error to size normalization for the total volume.
4. Computation of the errors of the total volume estimations (and associated correction factors).
5. Main experimental results: total volume estimations, with the proposed extraction of the copepod's prosome and the proposed corrections factors.
6. Study on the robustness of the results with respect to the simulation parameters.

**Chapter 8 – Correcting Total Copepods' Volume Estimates:**

8.1	Total volume correction . . . . .	84
8.1.1	Proposed approach . . . . .	84
8.1.2	Invariance of total volume estimation error to size normalization . . .	85
8.2	Experimental results with the UVP5-Cop dataset . . . . .	87
8.3	Robustness of the method . . . . .	88
8.3.1	Shape . . . . .	88
8.3.2	Orientation . . . . .	89
8.3.3	Number of simulated ellipsoids . . . . .	90
8.4	Discussion . . . . .	90

**8.1 Total volume correction**

Instead of proposing a novel volume estimation method, our approach is to study the errors made by the standard methods  $\mathcal{M}_{\text{ESD}}$  or  $\mathcal{M}_{\text{ELL}}$  in order to propose a procedure to compensate for these errors. Thus, the figures of past studies could be re-interpreted in light of the proposed corrections and marine ecologists could apply these corrections to future studies, sticking to their standard estimation method of choice.

**8.1.1 Proposed approach**

A set of  $N$  random ellipsoids with realistic proportions and various orientations is generated (as described in section 7.4), their projection silhouettes are computed following eq. (7.22), their volumes are estimated using  $\mathcal{M}_{\text{ESD}}$  (from the areas of the silhouettes) and  $\mathcal{M}_{\text{ELL}}$  (from the semi-axes of the silhouettes), and the error between the total, true, volume of all ellipsoids and the sum of the estimated volumes is computed as

$$\mathcal{T}_* = \frac{\sum_{n=1}^N V_*^n}{\sum_{n=1}^N V^n} = \frac{W_*}{W} \quad (8.1)$$

where the  $V^n$ s are the true volumes of the generated ellipsoids and the  $V_*^n$ s are the corresponding estimated volumes by method '\*' (ESD or ELL). Therefore,  $W$  is the true total volume of all ellipsoids in the simulation and  $W_*$  is the estimated total volume.

Once  $\mathcal{T}_*$  has been estimated from simulated ellipsoids, it can be used to correct the total volume estimated from  $P$  actual images of copepods as follows

$$W_*^c = \frac{\sum_{p=1}^P V_*^p}{\mathcal{T}_*} = \frac{W_*^u}{\mathcal{T}_*} \quad (8.2)$$

where  $V_*^p$  is the set of volumes estimated from the acquired images by method '\*',  $W_*^u$  is their total, and  $W_*^c$  is the corrected total estimated volume.

Note that the correction proposed in eq. (8.2) provides no objective element to prefer  $\mathcal{M}_{\text{ESD}}$  over  $\mathcal{M}_{\text{ELL}}$ , or vice versa. Indeed, the respective correction factors theoretically allow to perfectly retrieve the true total volume. In practice, though, the  $\mathcal{M}_{\text{ESD}}$  method might be a better option since the area measurement it relies on is more robust (*i.e.*, less sensitive to acquisition noise and greyscale variations) than the ellipse fit performed in the  $\mathcal{M}_{\text{ELL}}$  method.

An algorithmic description of the proposed method is given in appendix E. An implementation in *python* is available at the Inria GitLab<sup>1</sup>.

<sup>1</sup><https://gitlab.inria.fr/cedubois/Copepod-Volume-Correction>

### 8.1.2 Invariance of total volume estimation error to size normalization

As mentioned earlier, in section 7.4, we generated ellipsoids normalized by their size (*i.e.*, major semi-axis  $r_1$ ). Until now, it was assumed that the total volume errors (and so corrections) were invariant to this scaling. Here we present an empirical result that shows that it is indeed a reasonable assumption.

Let  $V^i, i \in [1..n]$ , be a set of true ellipsoid volumes, let  $V_*^i$  be the corresponding estimated volumes by the method ‘\*’ and let  $\mathcal{E}_*^i$  be the associated individual volume estimation errors. The total volume estimation error is

$$\mathcal{T}_* = \frac{\sum_i V_*^i}{\sum_i V^i} = \frac{\bar{V}_*}{\bar{V}} \quad (8.3)$$

where  $\bar{X}$  denotes the average of  $X$ . Now, suppose that each ellipsoid volume is scaled by a factor  $\alpha_i$  ( $U^i = \alpha_i V^i$ ), for example as a result of the normalization of the ellipsoid’s sizes by dividing their semi-axes  $r_1^i, r_2^i$ , and  $r_3^i$  by  $r_1^i$ . How will the estimated volumes  $U_*^i$  vary with respect to  $V_*^i$ ? From section 7.3.2, we know that  $\mathcal{E}_*^i = V_*^i/V^i$  is invariant to ellipsoid scaling. Therefore,  $U_*^i/U^i$  must still be equal to  $\mathcal{E}_*^i$ , which implies that  $U_*^i = \alpha_i V_*^i$ . Hence, the total volume estimation error after scaling is

$$\mathcal{T}'_* = \frac{\sum_i U_*^i}{\sum_i U^i} = \frac{\sum_i \alpha_i V_*^i}{\sum_i \alpha_i V^i} = \frac{\overline{\alpha V_*}}{\overline{\alpha V}}, \quad (8.4)$$

which depends on the scaling. Nevertheless, it can be noted that for a constant value  $\alpha_i = \beta$ , eq. (8.4) and eq. (8.3) are equivalent (*i.e.*,  $\mathcal{T}'_* = \mathcal{T}_*$ ). Considering a non-constant  $\alpha_i$ , the deviation between both equations might be more or less negligible. However, for the specific scaling  $\alpha = 1/r_1^3$  (size normalization), it can be observed empirically that the ratio  $\mathcal{T}'_*/\mathcal{T}_*$  is almost equal to one for various PDFs of  $r_1$  (see table 8.1 and figs. 8.1 and 8.2). It illustrates that the deviation is negligible, hence, validating the approximation

$$\mathcal{T}'_* = \frac{\overline{\alpha V_*}}{\overline{\alpha V}} \simeq \frac{\bar{V}_*}{\bar{V}} = \mathcal{T}_*. \quad (8.5)$$

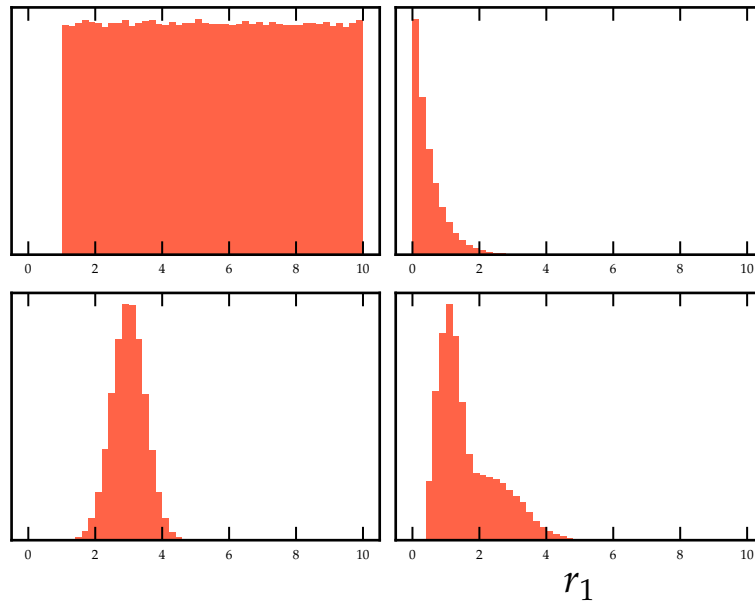
$r_1$ PDF	$\mathcal{T}'_{\text{ESD}}/\mathcal{T}_{\text{ESD}}$ [%]	$\mathcal{T}'_{\text{ELL}}/\mathcal{T}_{\text{ELL}}$ [%]
Uniform [1, 10]	99.971	99.964
Exponential $\lambda = 0.5$	99.945	99.954
Normal $\mu = 3, \sigma = 0.5$	100.002	100.015
$r_1$ KDE	99.966	99.988

**Table 8.1** The normalization factor for  $\mathcal{T}'_*$  is equal to  $\alpha = 1/r_1^3$ . For all experiments, the number of ellipsoids is  $N = 10^6$  and  $r_1 \geq r_2 \geq r_3 > 0$ . The ‘ $r_1$  KDE’ is the KDE [Parzen, 1962; Scott, 1979] obtained with the values of  $r_1$  measured on the images of copepods labelled as *top/bottom* or *side*. See figs. 8.1 and 8.2 for the illustrations of the PDFs.

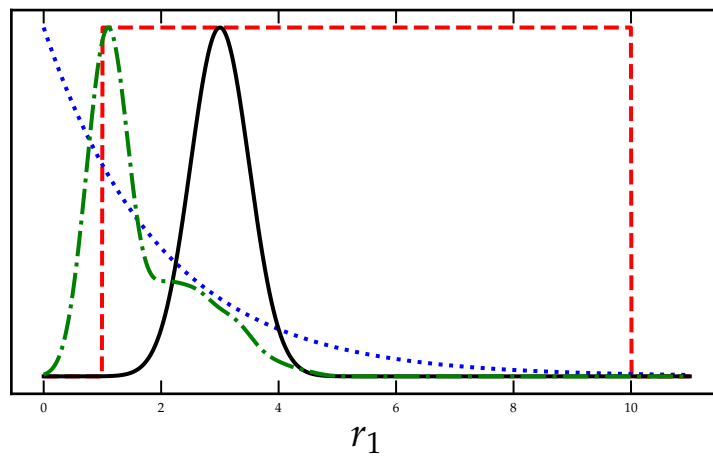
An intuition for explaining the validity of this approximation is that  $V_*$  and  $V$  are correlated, since the estimated volumes are expected to ‘follow’ the real volumes. An extreme position would be to consider a linear dependency  $V_* = \beta V$ . In this case, eq. (8.5) is effectively an equality. Then, if their correlation is high enough, eq. (8.5) should still be an appropriate approximation.

To conclude, the error on the total volume can safely be considered invariant to size normalization. As a consequence, the randomly generated ellipsoids used to determine the total volume correction factor (see sections 7.4 and 8.1 and, in particular, eq. (8.1)) can be generated with a constant  $r_1 = 1$  and using only the PDFs of the ratios  $r_2/r_1$  and  $r_3/r_1$ , like before (section 7.4.2).





**Figure 8.1** Histograms of  $r_1$  samples (normalized by their maximum) corresponding to the PDFs used in the experiments (see table 8.1). Top left: Uniform; top right: Exponential; bottom left: Normal; bottom right: KDE.



**Figure 8.2** Schematic view of the  $r_1$  PDFs (normalized by their maximum) used in the experiments (see table 8.1).

Let us remark that (i) it allows performing the simulation in a 2-d space only, and (ii) the computation of the correction factor does not require any assumptions on the  $r_1$  PDF. This last point is crucial, meaning that the correction factor for the total volume can be computed and applied on data sets with any  $r_1$  distributions, as long as the copepods proportions (*i.e.*, axes ratios) remain similar.

## 8.2 Experimental results with the UVP5-Cop dataset

With the axes ratio PDFs fitted to the manually selected data (section 7.4.2), we followed the previous simulation procedure to generate a set of realistic ellipsoids, compute their projection, compute their true and estimated volumes and finally compute the error of the total estimations  $\mathcal{T}_*$ , for both methods. We generated  $10^8$  ellipsoids in order to cover the shape and orientation parameters space with a high enough resolution (see section 8.3 for a study of the influence of the number of ellipsoids). Using eq. (8.1), we obtained  $\mathcal{T}_{\text{ESD}} = 122\%$  and  $\mathcal{T}_{\text{ELL}} = 87\%$ . In other words, on average,  $\mathcal{M}_{\text{ESD}}$  overestimated the true volume by 22% while  $\mathcal{M}_{\text{ELL}}$  underestimated it by 13%.

Let us note that, as expected,  $\mathcal{T}_{\text{ESD}}$  is higher than  $\mathcal{T}_{\text{ELL}}$ , since we showed that the  $\mathcal{M}_{\text{ESD}}$  volume is always greater than or equal to the  $\mathcal{M}_{\text{ELL}}$  volume (see section 5.5.2). Also, the correction factors are simply computed as  $1/\mathcal{T}_{\text{ESD}}$  and  $1/\mathcal{T}_{\text{ELL}}$ .

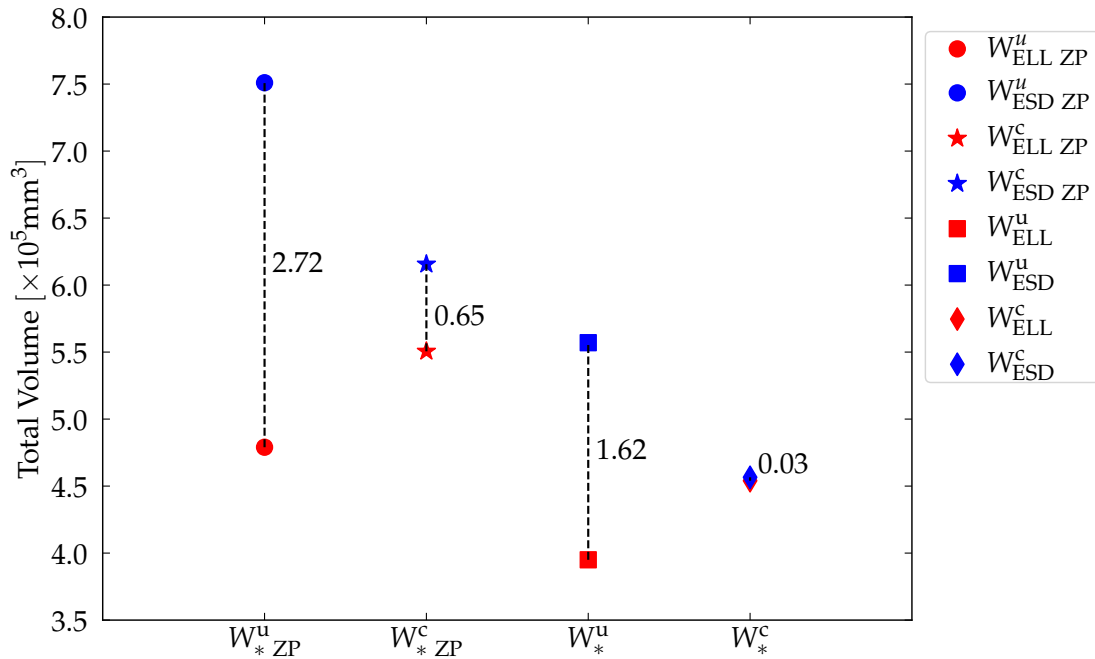
As a main experimental result of this part of the thesis, we estimated the total volume of the  $\sim 150\text{k}$  copepods in the UVP5-Cop data set using a variety of approaches: (i) the  $\mathcal{M}_{\text{ESD}}$  or  $\mathcal{M}_{\text{ELL}}$  method as computed originally by ZooProcess (noted ZP here), with an ellipse fit based on the image mask, or using our improved method based on the mask of the copepod's body only (chapter 6), and (ii) before ( $W^u$ ) and after ( $W^c$ ) correction by the factors defined above. This produce a total of eight estimations, presented in fig. 8.3 and table 8.2.

Method	$W_{\text{ESD}} [\times 10^5 \text{mm}^3]$	$W_{\text{ELL}} [\times 10^5 \text{mm}^3]$	Gap $[\times 10^5 \text{mm}^3]$
Zooprocess ( $W^u_{* \text{ZP}}$ )	7.51	4.79	2.72
Zooprocess & correction ( $W^c_{* \text{ZP}}$ )	6.16	5.51	0.65
Proposed fit ( $W^u_{*}$ )	5.57	3.95	1.62
Proposed fit & correction ( $W^c_{*}$ )	4.57	4.54	0.03

**Table 8.2** Overview of the total volumes estimations. The results of the two first rows were computed with the State-Of-The-Art area estimation and ellipse fit by ZooProcess (ZP). The two last ones were computed using the new procedure presented in chapter 6. The second and fourth rows are results after applying the correction factors ( $W^c$ ). The *Gap* column represents the absolute difference between the total volume estimations with  $\mathcal{M}_{\text{ESD}}$  and  $\mathcal{M}_{\text{ELL}}$ .

In theory (*i.e.*, copepods are ellipsoids, the imaging system has perfect lenses and infinite resolution, surface measurements and ellipse fits are exact, the simulation parameters match the reality, an infinite number of samples are generated), we should obtain exactly the same total volume estimations with  $\mathcal{M}_{\text{ESD}}$  and  $\mathcal{M}_{\text{ELL}}$  after correction. Naturally, this is not the case in practice, but we can assess the effectiveness of the proposed volume correction by checking how the discrepancy between the  $\mathcal{M}_{\text{ESD}}$  or  $\mathcal{M}_{\text{ELL}}$  estimations decreases after applying the correction. This gap is divided by 4 when using the ZooProcess measurements and by 54 when using our improved versions. The fact that the corrected volumes ( $W^c_{\text{ESD}}$  and  $W^c_{\text{ELL}}$ ) seem to converge is no proof that either one is the truth, but it at least suggests that the proposed correction method brings a significant improvement.

The effect of the improved area and ellipse measurement alone can also be gauged in the same way. The discrepancy is divided by 2 when comparing  $W^u_{* \text{ZP}}$  and  $W^u_{*}$  (as seen in chapter 6),



**Figure 8.3** Total volume estimated by  $\mathcal{M}_{ESD}$  (blue) or  $\mathcal{M}_{ELL}$  (red) obtained by ZooProcess (ZP) and our improved measures, uncorrected ( $W^u$ ) or corrected ( $W^c$ ). The discrepancies between the methods are highlighted by dashed lines and annotated by the corresponding absolute differences. The proposed correction drastically reduces these discrepancies.

and by 22 when comparing the corrected versions,  $W_{*}^c_{ZP}$  and  $W_{*}^c$ . Overall, if we compare the current state of the art (uncorrected total volumes obtained using ZooProcess) and the corrected total volumes obtained using improved image processing, the discrepancy is divided by 91, bringing the  $\mathcal{M}_{ESD}$  and  $\mathcal{M}_{ELL}$  estimations very close to each other.

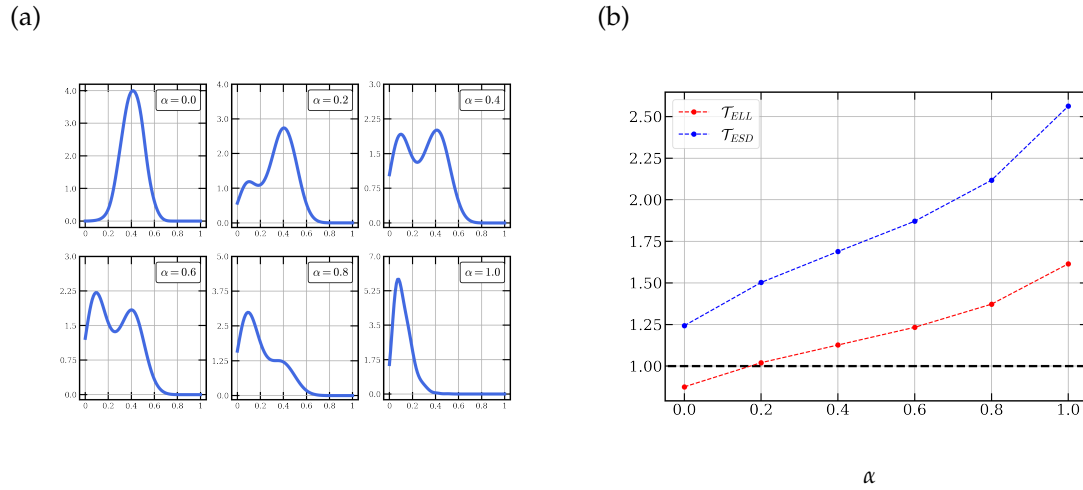
### 8.3 Robustness of the method

This section aims at testing the robustness of the proposed method with respect to its parameters, in the context of the previous experiment, *i.e.*, with manual shapes measurements from the UVP5-Cop data set and a uniform random ellipsoid orientation. For each experiment, we let one of the parameters free to observe its influence on the correction factors.

#### 8.3.1 Shape

In our data set, the distribution of semi-axes ratios was unimodal (see fig. 7.2). Nevertheless, the proposed method can accommodate any distribution thanks to the use of the KDE approach [Parzen, 1962; Scott, 1979]. Thus, it is interesting to verify how the correction factors vary in a multimodal scenario, for example when two populations of copepods with different shapes are present. Figure 8.4 shows some examples of synthetic PDFs of  $r_3/r_1$  for a mixture of two body shape distributions with varying proportions, and the effect on the correction factors, the other distributions ( $r_2/r_1$  and body orientation) being fixed. The top-left panel represents organisms with a round cross-section, that is to say  $r_3 \sim r_2$  (*i.e.*, resembling copepods of the order Calanoida). Therefore, we used the same distributions for  $r_3/r_1$  and  $r_2/r_1$ : a Normal law with

mean 0.41 and variance 0.0076 (values based on the fit to the UVP5-Cop data). The bottom-right panel represents organisms with a flatter cross-section *i.e.*,  $r_3 \ll r_2$  (*i.e.*, like the copepods in the order Harpacticoida). We used a Beta distribution of parameters  $a = 2$  and  $b = 15$ . The parameter  $\alpha \in [0, 1]$  determines the proportions of samples from the two populations ( $\alpha = 0$ : 100% of the samples are from the first population;  $\alpha = 1$ : 100% are from the second one). The computed correction factors increase with  $\alpha$ , *i.e.*, as the proportion of flatter organisms increases. This is to be expected since the viewing angle has more consequences on the appearance of the projection of the organism for these flattened shapes. This illustrates that the correction factors strongly depend on community composition (and therefore on a correct modelling of the data). In particular, the correction factors obtained in section 8.2 for the global UVP5-Cop data set should not be used blindly on other data sets. Instead, the required PDFs should be estimated from the data and the correction factors recomputed.



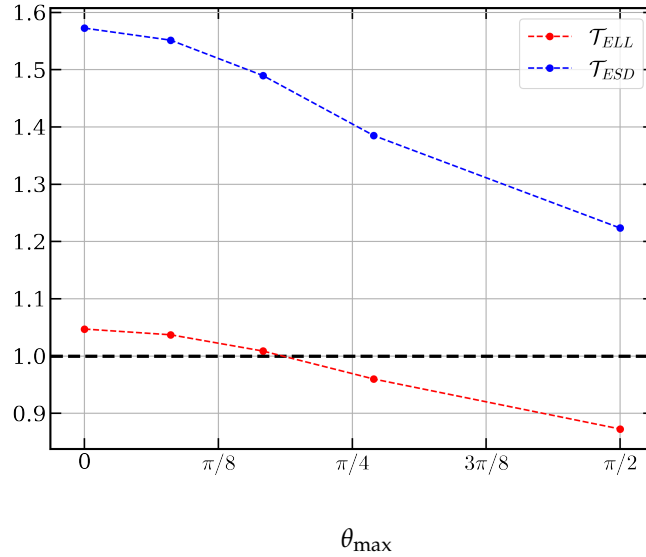
**Figure 8.4** (a) Distribution of  $r_3/r_1$  for various mixtures of two subpopulations, from  $\alpha = 0$  (Normal distribution fitted on the  $r_2/r_1$  data of the UVP5-Cop data set), to  $\alpha = 1$ , Beta distribution with parameters  $a = 2$  and  $b = 15$ . (b) The corresponding correction factors for  $N = 10^6$  ellipsoids. The black dashed line indicates a correction factor of one, *i.e.*, no error.

### 8.3.2 Orientation

In our simulations, the orientation of copepods relative to the camera was considered uniformly random. Nevertheless, with other imaging instruments, the orientation may not be uniformly distributed (*e.g.*, with scanners like the ZooScan or in-flow imagers such as the FlowCam, the orientation of organisms relative to the imaging sensor is mechanically constrained). It is possible to relax the uniformity assumption and check how the correction factors vary with different degrees of constraint on the orientation.

Rotation can be performed around the x-axis of the copepod/ellipsoid (*i.e.*, the length) and determines whether we get a dorsal, side or ventral view (or something in-between); along the y-axis (*i.e.*, the width) and, in the case of the UVP5, this changes the vertical tilt of the organism; and along the z-axis, normal to the view plane, which, in the case of the UVP5, changes the ‘cardinal’ orientation of the organism. In the simulations, rotation along the z-axis is set to the identity (*i.e.*, no rotation) since this does not influence the results at all; the x-axis rotation is free and uniform in  $[0, \pi]$  (so that  $\rho_2 \in [r_3, r_2]$ ); and the y-axis rotation is uniform in the interval

$[0, \theta_{\max}]$ : the higher  $\theta_{\max}$ , the more the ellipsoid can rotate vertically. Figure 8.5 shows the correction factors obtained. When  $\theta_{\max} = 0$ , all copepods are aligned on a plane (a 'ZooScan-like'



**Figure 8.5** Evolution of the correction factors for varying ranges of allowed vertical rotation. The rotation angle around the y-axis,  $\theta$ , is restricted to the interval  $[0, \theta_{\max}]$ . Each dot of the plots corresponds to a simulation for a particular  $\theta_{\max}$ . The black dashed line indicates a correction factor of one, *i.e.*, no error.

scenario). When  $\theta_{\max} = \pi/2$ , the results are the same as with a random, uniform distribution. It is interesting to note that: (i) the correction factor for the  $\mathcal{M}_{ELL}$  method is relatively stable, while it varies more significantly for the  $\mathcal{M}_{ESD}$  method; (ii) the variation among the different orientation scenarios is much lower than for the different shapes (Figure 8.4).

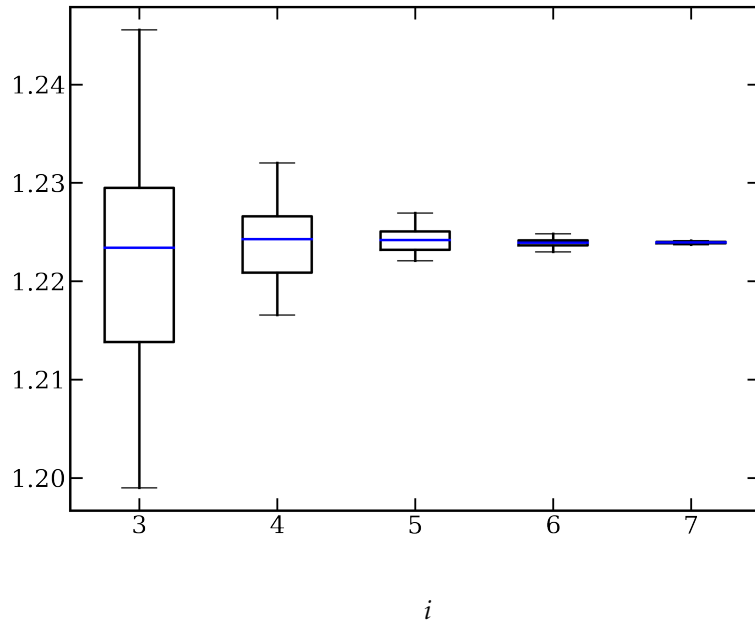
### 8.3.3 Number of simulated ellipsoids

The number of ellipsoids generated in the simulation ( $N$ ) only determines the precision of the estimation of the correction factors. As a matter of fact, the estimation of the correction factors tends to be perfect when  $N$  tends towards infinity. But the value of  $N$  still influences the duration of simulations and the computing power required, so it is interesting to get an idea of its influence on the variance of the computed factors. Thus, we performed several simulations with various numbers of ellipsoids  $N_i = 10^i, i \in \{3, 4, 5, 6, 7\}$ , lower than  $N = 10^8$  used in section 8.2 (50 times for each  $i$ ). Figure 8.6 shows that the variance of the correction factors becomes negligible for  $N_i \geq 10^6$ .

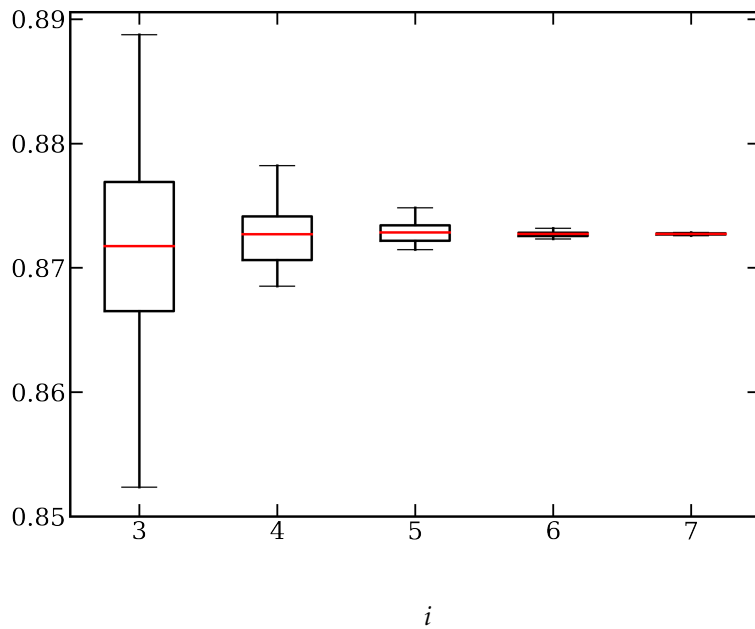
## 8.4 Discussion

This chapter described an application to the UVP5-Cop data set of the procedure presented in chapter 7 for correcting the error (due to the 3-d to 2-d projection) on the estimation of the total volume of copepods. A potential weakness of this application is that the distributions of semi-axes ratios were estimated from a relatively small number of images (<300 for each). Since identifying copepods in a given orientation is very time-consuming, a useful alternative would

(a)



(b)



**Figure 8.6** Corrections factors ((a)  $\mathcal{M}_{\text{ESD}}$ , (b)  $\mathcal{M}_{\text{ELL}}$ ) for  $N_i = 10^i, i \in \{3, 4, 5, 6, 7\}$ . For each  $i$ , we computed the correction factors 50 times. The blue (red) line is the mean factor for the  $\mathcal{M}_{\text{ESD}}$  ( $\mathcal{M}_{\text{ELL}}$ ) method. The boxes and whiskers are drawn according to Tukey's definition [McGill et al., 1978].

be to use a classifier that could automatically identify copepods seen from above or from the side within the total data set. Copepods were isolated from other organisms through a combination of machine learning and human classification. The same tools were used to optimize a custom classifier for side *vs.* top/bottom *vs.* other angle copepods. While it accelerated the collection of the examples in the sample sets, it did not achieve great accuracy, largely because of the overwhelming dominance of copepods seen from 'other angles'.

The other assumption was in the choice of a uniform orientation distribution. While copepods in a given environment may orient themselves in a particular manner, vertically towards the surface for example [Benfield et al., 2000], quantitative information on such behaviour is very scarce. However, it is very likely to change with location, depth, time of day, organism age, condition, etc. Since our data set contains >150k organisms of various sizes, from different locations, depths and dates, assuming a uniform distribution overall was the only possible choice and likely reflects the reality. Still, to gauge the influence of the orientation distribution for applications to other, more restricted, data sets, we designed simulations in a non-uniform scenario and the influence on the correction factor proved to be limited (see section 8.3.2). *In situ* instruments that do not disturb the water and image in three dimensions (*e.g.*, through holography) are the only viable solution to yield quantitative information on orientation, from which an estimation of the PDF of the orientation angles could be performed. Such instruments are, unfortunately, very scarcely used. Other alternatives would be to use realistic 3-d models of copepods, generate 2-d views from them, and optimize either an image-to-orientation regressor (to directly predict the orientation) or an augmented auto-encoder convolutional neural network (to access the orientation encoded in the central part of the network, *e.g.*, see Sundermeyer et al. [2018]). Nevertheless, those 'learning-based' methods strongly rely on a 'training' data set of synthetic images, and, by extension, on prior knowledge on the organisms' shape and on the acquisition system, many of which are not available for plankton imaging instruments.

## Chapter 9

### Insight on the error on Normalized Bio-volume Size Spectra

#### **Key points – NBSS of simulated volumes of copepods**

1. The NBSS is based on volume estimations.
2. We discuss the influence of the organism's projection for NBSS measurements.

#### **Contributions**

3. Estimation of the distribution of the major semi-axis  $r_1$ .
4. Simulation of absolute-size ellipsoids.
5. Error on the estimated NBSSs.



**Chapter 9 – Insight on the error on Normalized Bio-volume Size Spectra:**

9.1	Introduction . . . . .	94
9.2	Method for estimating the major semi-axes $r_1$ distribution . . . . .	94
9.3	Experiment . . . . .	95

**9.1 Introduction**

Plankton images are often used to estimate the size of organisms and, in particular, to compute Normalized Bio-volume Size Spectra (NBSS). It is the histogram of the logarithm of the volumes, and each count is normalized by the width of its bin [Kerr and Dickie, 2001]. The slope of these spectra is a proxy for the efficacy of the energy transfer from small to large organisms within an ecosystem [Sprules and Barth, 2016]. An NBSS obviously depends on the measurement of the volume of the organisms it encompasses and, therefore, its slope may be affected by the error in the volume estimation due to the 3-d to 2-d projection. We will test this through a simulation procedure, similar to the one described in chapter 7, and compare the NBSS and the slope values computed from the volumes of randomly generated ellipsoids (considered as the true value) to that derived from estimations of volume from their projections using  $\mathcal{M}_{\text{ESD}}$  and  $\mathcal{M}_{\text{ELL}}$ .

Since the NBSS depends on absolute sizes, the simulator based on size-normalized ellipsoids, defined by  $r_1 = 1$  and two semi-axes ratios (see details in chapters 7 and 8), must be modified to use directly  $r_1$ ,  $r_2$ , and  $r_3$ , although this is a less favourable statistical context (see section section 7.4). As mentioned previously, the distributions of  $r_1$ ,  $r_2$ , and  $r_3$  can be defined from the literature or estimated from measurements.

**9.2 Method for estimating the major semi-axes  $r_1$  distribution**

As explained above, computing a NBSS relies on absolute volumes and the distributions of the three semi-axes ( $r_1$ ,  $r_2$ , and  $r_3$ ) are the main parameters. They could have been estimated directly on the 295 + 265 samples used to estimate the distributions of ratios above. However, the distribution of  $r_1$  for the 295 copepods viewed from the side shows a bias towards larger sizes, because it is difficult to tell whether a copepod is indeed viewed from its side when it is small (see appendix F). This bias has limited influence on the estimation of the distributions of the ratios,  $r_2/r_1$  and  $r_3/r_1$  (used in chapters 7 and 8), since the both relationships  $r_1$  vs.  $r_2$  and  $r_1$  vs.  $r_3$  are fairly linear (fig. F.2). Here, we propose a method for estimating the distribution of  $r_1$  that relies on one strong hypothesis: *the expectation for the major semi-axis distribution is an exponential decay*  $P(r_1) = \lambda \exp(-\lambda r_1)$  (see Sprules and Barth [2016]). The simulation of an ellipsoid now relies on the observation of  $r_2$  and  $r_3$ , and on the inference of the  $r_1$  distribution (parametrized by parameter  $\lambda$ ). With the simulator and the volume data at hand, *i.e.*, the volumes measurements with  $\mathcal{M}_{\text{ESD}}$  and  $\mathcal{M}_{\text{ELL}}$  on the UVP5-Cop images, we propose to define  $\lambda$  by comparing the observed volume estimations  $W_\star^u$  with the simulated ones, noted  $W_\star^s(\lambda)$  (here  $\star$  is either ESD or ELL). We use an absolute difference of their mean, one for each estimation method. Hence, we search for the exponential law of parameter  $\lambda$  that minimize the absolute difference of the estimated volumes (observed and simulated)

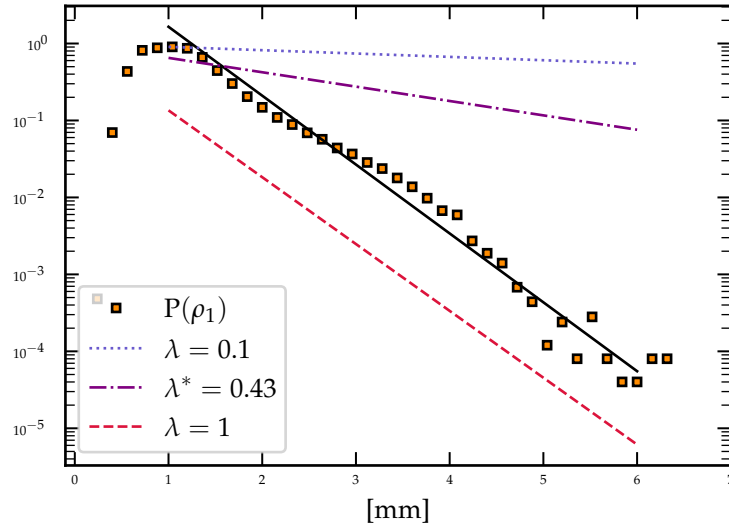
$$\lambda^* = \arg \min_{\lambda} \left| \frac{1}{P} W_\star^u - \frac{1}{N} W_\star^s(\lambda) \right|, \quad (9.1)$$

with  $P$  the number of copepods images (observations) and  $N$  the number of simulated ellipsoids. Note that, in this chapter, the subscript  $\star$  denotes the use of either  $\mathcal{M}_{\text{ESD}}$  or  $\mathcal{M}_{\text{ELL}}$ , while the superscript  $\star$  denotes optimality.

### 9.3 Experiment

With 100 test-values of  $\lambda$ , we compute the mean volume across  $N = 10^5$  simulated ellipsoids, as well as the estimations with both  $\mathcal{M}_{\text{ESD}}$  and  $\mathcal{M}_{\text{ELL}}$  from the projections of the ellipsoids. For each test-value of  $\lambda$ , the major semi-axes  $r_1$  are drawn from the corresponding exponential distribution (note, the minimum value of  $r_1$  is set to 1 mm, *i.e.*, the distribution is truncated according to the detection limit of the UVP5 camera<sup>1</sup>) and the others axes ratios are first sampled according to the previous procedure (axes ratios in chapter 7) and then multiplied by the corresponding (just drawn) semi-major axis  $r_1$ .

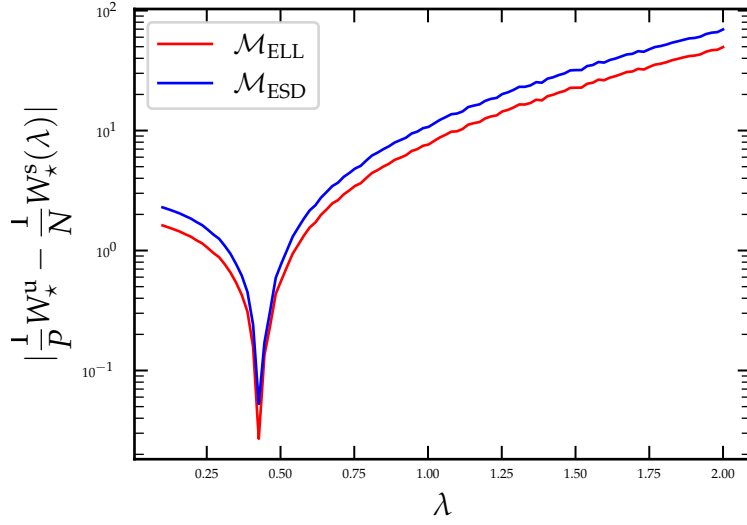
We computed the absolute difference for 100 different  $\lambda$  values, defined with a linear range. The extremal values are  $\lambda = 0.1$  and  $\lambda = 2$ . In logarithm scale, the exponential decay is of affine form  $-\lambda r_1 + \log \lambda$ , with  $-\lambda$  the slope and  $\log \lambda$  the intercept. A representation of the slopes is shown in fig. 9.1 for the extremal values and  $\lambda^*$ , the one that minimizes the absolute difference among (eq. (9.1)) for both  $\mathcal{M}_{\text{ESD}}$  and  $\mathcal{M}_{\text{ELL}}$ , among the tested values. The black solid line shows the fit of the histogram of the projected semi-axis  $P(\rho_1)$  measured from the images (orange squares), as a reference. The slope of the distribution of  $r_1$  is expected to be less steep than the one of  $\rho_1$  since  $\rho_1 \leq r_1$ . The evolution of the absolute differences (for both  $\mathcal{M}_{\text{ESD}}$  and  $\mathcal{M}_{\text{ELL}}$ ) are given in fig. 9.2. The tested value that minimize the absolute difference (eq. (9.1)) is  $\lambda^* = 0.43$ , for both methods. Note that, even if the figure suggests that  $\lambda^* = 0.43$  is a reasonable choice, there is no evidence that eq. (9.1) have a unique solution.



**Figure 9.1** Histogram of the major semi-axes  $\rho_1$  (measured as in chapter 6) of the UVP5-Cop dataset, represented as orange squares. The coloured dashed, dotted or both are representations of the exponential distributions of  $r_1$  for the minimum, the maximum and the optimal parameter (see legend); only the slope is meaningful here, the intercepts have been chosen to improve the reading. Note the logarithmic scale. The x-axis is the length of  $\rho_1$  and  $r_1$ .

A set of  $10^7$  ellipsoids was generated with the optimal value  $\lambda^*$  (*i.e.*, values of  $r_1$  are drawn from an exponential distribution of parameter  $\lambda^* = 0.43$ ). The ellipsoid volumes are used to compute the simulated ground-truth NBSS. Each ellipsoid is also projected, its volume is

<sup>1</sup>the detection limit is  $\sim 1$  mm, which explain the decrease of the histogram for  $\rho_1 < 1$  mm in fig. 9.1

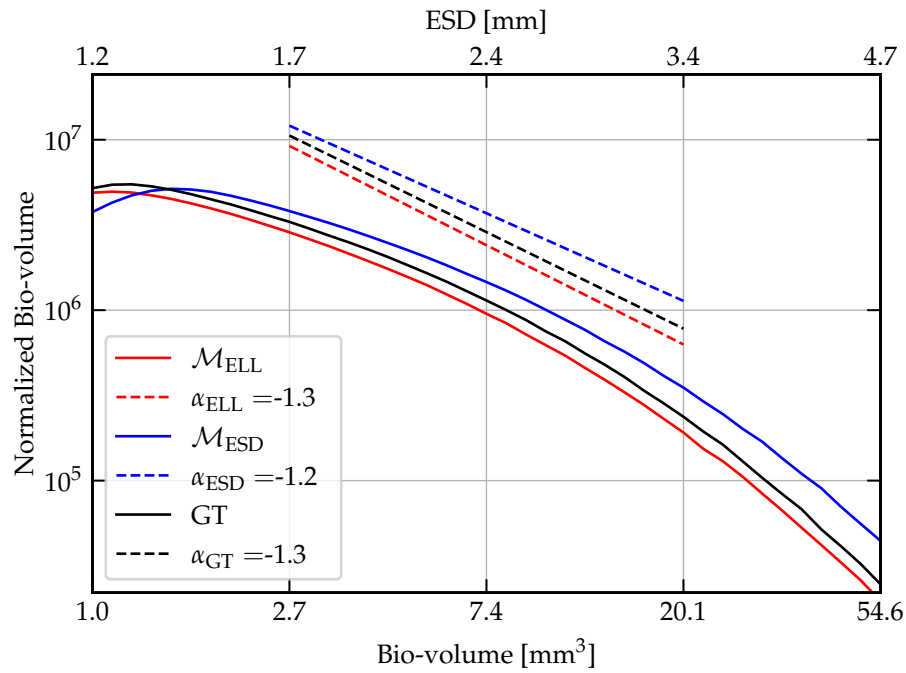


**Figure 9.2** Absolute difference between the measured mean volume  $\frac{1}{P} W_*^u$  and  $\frac{1}{N} W_*^s$  for both estimation methods. Note the logarithmic scale.

estimated with both  $\mathcal{M}_{\text{ESD}}$  and  $\mathcal{M}_{\text{ELL}}$ , and the corresponding NBSS are computed (see fig. 9.3). In all three cases, the bin size is  $0.1 \text{ mm}^3$  in log space. To assess whether the two volume estimation methods influence the estimation of the efficacy of the energy transfer in ecosystems, the slopes of a linear fit to each NBSS in the interval  $[2.7, 20.1] \text{ mm}^3$  (or  $[1.7, 3.4] \text{ mm ESD}$ ) were computed (see fig. 9.3).

The ground-truth NBSS was between the  $\mathcal{M}_{\text{ELL}}$  and  $\mathcal{M}_{\text{ESD}}$  estimations. More importantly, the three slopes were very close to each other:  $-1.3$  for the ground-truth *vs.*  $-1.2$  for  $\mathcal{M}_{\text{ESD}}$  and  $-1.3$  for  $\mathcal{M}_{\text{ELL}}$ . To compare this with the range of natural variability in the data set, we computed the NBSS from images collected in polar (absolute value of latitude in  $[60^\circ, 90^\circ]$ ) and temperate (absolute value of the latitude in  $[20^\circ, 40^\circ]$ ) regions, between 0 and 150 m depth. The mode of energy transfer is expected to be very different between these two ecosystems, and indeed, the slopes of these NBSS were  $-0.7$  (polar) and  $-1.4$  (temperate) with  $\mathcal{M}_{\text{ESD}}$  and  $-0.7$  and  $-1.6$  with  $\mathcal{M}_{\text{ELL}}$ . The amplitude of natural variability is much larger than the variability induced by the volume estimation method. Therefore, despite the errors the estimation methods induce on individual volume (fig. 5.5), both  $\mathcal{M}_{\text{ESD}}$  and  $\mathcal{M}_{\text{ELL}}$  seem to be valid approaches to compute the NBSS and to infer the energy transfer efficiency through a linear fit. Let us remark that, if a choice between both methods was necessary, the NBSS estimated with  $\mathcal{M}_{\text{ELL}}$  would be preferable since it follows the ground-truth NBSS slightly better (see fig. 9.3).

The experiment presented here is based on the simulation of ellipsoids and their projections (as defined in chapter 7). A main difference with the application in chapter 8 is the relaxation of the size normalization. This allows to access to the distribution of the absolute (as opposed to size normalized) simulated volumes, which is necessary to compute the NBSS. In addition, it can be used to estimate the mean individual copepod volume (according to the simulation parameters) which provides a new total volume estimation method, by simply multiplying the mean value with the observed absolute number of copepods. The mean volume obtained from the simulation used in this chapter is  $\bar{V}=2.88 \text{ mm}^3$ , it corresponds to a total of  $W = 4.49 \times 10^5 \text{ mm}^3$  for 155,945 copepods, which is in line with the previous results in chapter 8 ( $4.57 \times 10^5$



**Figure 9.3** Simulated NBSSs: ground-truth (GT) in black; estimation from  $\mathcal{M}_{ELL}$  (resp.  $\mathcal{M}_{ESD}$ ) in red (resp. blue). The x-axis is given in volume (mm<sup>3</sup>) but also in ESD (mm) for comparability with other work. The dashed lines show the linear fits (which are offset vertically for improved readability).

for  $\mathcal{M}_{ESD}$  and  $4.54 \times 10^5$  for  $\mathcal{M}_{ELL}$ ). Nevertheless, for the total volume estimation, the previous method is preferred since it relies only on two distributions of the axis-ratios rather than three distributions of the absolute axes here.



## Chapter 10

### Conclusion

The study of the plankton organisms is of primary interest for our understanding of marine ecosystems machinery and the large biogeochemical fluxes in the ocean. The organisms move with the water mass in which they are embedded in, making them accurate markers of local aquatic environments. They are relatively small (even if their size spectrum is wide) but extremely abundant. Hence, they feed the ocean animals and the sum of their individual contributions have a strong impact on the climate regulation, through the biological carbon pump. Plankton ecology aims at studying their interactions, with themselves and with their environment. For this purpose, the work of marine ecologists partly rely on the observation of the planktonic organisms. Those are not visible to the naked eye (at least the majority of them). This is why the developments of digital imaging systems (from microscope in the lab to specific video cameras in the ocean), has been a tipping point in the understanding of the marine ecosystems. It facilitated the taxonomic classification and led to the discovery of new taxa. Moreover, it paved the way for the study of functional traits, among which the size stands out. In particular, the elaboration of *in situ* imaging instruments enabled to shed light on them directly in their environment. Such technical improvements led to the acquisition of millions of images across the world. Their manual analysis is time-consuming due to their number. Therefore, the development of automated image processing methods has been critical. Although such approaches could be much faster than a human relying on the computational power of computers, they have limited performances compared to the human intelligence. In particular, the classification of the organism images is a challenging task. The introduction of the Artificial Neural Networks (ANNs) and, in particular, the Convolutional Neural Networks (CNNs), has been a breakthrough for this purpose. In particular, their optimization using directly the images, as opposed to hand-crafted image features, is a major strength. Those facilitate the identification of rare plankton taxa, bringing the classification of plankton images to unprecedented levels of accuracy. On the other hand, the predictions of such methods are not yet clearly explainable. This can be restrictive in practice, since ecologists need reliable and trustworthy models. They also use standard classification models on image features extracted with CNNs in practice. This procedure takes advantages of both side as it is based on optimized features and trustable classifiers.

In the first part of this thesis, we proposed a method of classification inspired by a ANN procedure, but with a simple and interpretable sample transformation. First, we noted that the classification decision of ANNs is given by the Nearest Target (NT) classifier on the transformed samples, with one target per class. Hence, we studied the influence of the position of the target in our simple framework and conjectured that taking equidistant targets fulfilled an optimality criterion. Second, we demonstrated that the resulting classifier with this choice of targets is a Weigthed-Nearest-Neighbours (WNN) classifier. From this result, we highlighted the existence of a kernel associated to the WNN, that we called a 'nearest-neighbour kernel'. The existence of

an optimal targets choice in the sense of the misclassification risk is rather unclear. Alternatively, one would consider the imbalance of the number of samples per class in our development. In practice, we proposed a modern implementation of the WNN classifier, and we show it can produce accurate results on two reference plankton images data sets, one of collected samples and another, more challenging, of *in situ* observations. We highlighted the advantages and limitations of the proposed implementation compared to a classifier commonly used by the community, namely a Random Forest (RF). Additionally, we showed that the proposed kernel can be used with linear classifiers as Support Vector Machines (SVMs). Finally, other definitions of the sample weights might be explored, in particular inspired from dimension reduction techniques.

In the second part, we focused on one of the most abundant taxonomic group, the copepods. We tackled their volume estimations from *in situ* 2-d images. We highlighted the limitations of two standard methods used in the literature. Furthermore, we found two main sources of errors and proposed to correct them. First, the copepods antennas can affect the volume estimations. Hence, we proposed a method for selecting the copepods prosome only *i.e.*, without the antennas. Second, the projection of the prosome on the image plane make it impossible to estimate the true volume. We tackled the estimation of the total volume of the copepods based on an ellipsoidal model. With its exact projection onto a plane, we were able to simulate a set of realistic ellipsoids and their projection. From those, we measured the error made for the total volume estimation of the set with the two standard methods. The result is that one overestimate the total volume by  $\sim 20\%$  and the other underestimate it by  $\sim 10\%$ . Our result relies on two core parameters: the distributions of shape of the copepods and the distribution of their orientation. Concerning the former, we used manual measurements on a hundred of axis-aligned views of copepods. A major improvement would be to estimate the axis distributions on more data to have a more precise estimations of the distributions. Regarding the orientation, we made the hypothesis of a uniform distribution. In the absence of any priors on the orientation of the copepods at the global scale, this is the default assumptions *i.e.*, there is no clear evidence to put forward a particular orientation. Obviously, estimations of the orientation of the copepods at global scale could give a hint on our hypothesis relevance. Moreover, our model can adapt to other distributions. Hence, the estimation of the distribution of the orientation could be taken into account. As an application, from these simulated global errors, we derived two corrections factors and applied them to correct the total volume estimations of the copepods from images of the Underwater Vision Profiler 5 (UVP5) *in situ* camera. While we observed a significant decrease of the gap between the two standard estimations by applying the corrections, there is no evidence that the resulting estimation is close to the ground truth, as it is unknown. Therefore, the development of an experimental set-up with a ground truth would be benefic to assess the quality of correction method. Additionally, we showed that the simulator developed for the total volume correction can be used for other applications. In particular, it was used to show that the Normalized Bio-volume Size Spectra (NBSS) computed from individual volumes estimations on 2-d images is accurate for this data set.

To conclude, we addressed in this thesis the automatic classification of plankton images and the estimation of the total volume of copepods from *in situ* 2-d images. We developed new methods and demonstrated their performances in real applications cases.

## Appendix A

### Proposed classifier: development details

#### A.1 General notations

- The terms ‘positive’ and ‘negative’ are used in their strict, ‘not including zero’ sense. To include zero, ‘non-negative’ and ‘non-positive’ are used, respectively.
- $[1..n]$  is the set of integers from 1 to  $n$ .
- $\text{Card}(S)$  is the cardinal of the set  $S$ .
- $\mathbf{0}_d$  and  $\mathbf{1}_d$  are the vectors of zeros and ones in dimension  $d$ .
- $I_d$  is the  $d \times d$ -identity matrix.
- $v[i]$  is the  $i$ th component of the vector  $v$ .
- $M[i, j]$  is the element of the matrix  $M$  at the intersection between the  $i$ th row and the  $j$ th column. Replacing  $i$ , respectively  $j$ , with ‘:’ denotes the  $j$ th column, respectively  $i$ th row.
- $\text{Tr}(M)$  is the trace of the square matrix  $M$ .
- $\text{Vec}_{i=1}^d(\alpha_i)$  is the vector of components  $\alpha_i$  for  $i \in [1..d]$ .
- $\text{Diag}_{i=1}^d(\alpha_i)$  is the diagonal matrix obtained by placing  $\alpha_i$  along the diagonal for  $i \in [1..d]$ .
- $\text{Diag}_n(x)$  is the block diagonal matrix obtained by repeating  $n$  times  $x$  along the diagonal.
- $M^\square$  is the matrix used to build the block diagonal matrix  $M$ , that is  $M = \text{Diag}_n(M^\square)$ .



## A.2 Expression of $\mathcal{A}_{i,k}$

As a reminder (see eqs. (2.85) and (2.86)),

$$\begin{aligned} \mathcal{A}_{i,k} &= \sum_{l=1}^p (w_l^c(i))^2 |T_l - T_k|^2 \\ &\quad + 2 \sum_{l < m} w_l^c(i) w_m^c(i) (T_l - T_k) \cdot (T_m - T_k), \end{aligned} \tag{A.1}$$

and

$$(T_l - T_k) \cdot (T_m - T_k) = \begin{cases} \delta^2/2 & \text{if } l \neq k \text{ and } m \neq k \\ 0 & \text{otherwise} \end{cases}. \tag{A.2}$$

Then,

$$\begin{aligned} \mathcal{A}_{i,k} &= \delta^2 \sum_{\substack{l=1 \\ l \neq k}}^p (w_l^c(i))^2 \\ &\quad + \underbrace{\sum_{l=1}^p \sum_{\substack{m=1 \\ m \neq l}}^p w_l^c(i) w_m^c(i) (T_l - T_k) \cdot (T_m - T_k)}_{\mathcal{B}_{i,k}}, \end{aligned} \tag{A.3}$$

$$= \delta^2 \left( \sum_{l=1}^p (w_l^c(i))^2 - (w_k^c(i))^2 \right) + \mathcal{B}_{i,k}. \tag{A.4}$$

Next,

$$\mathcal{B}_{i,k} = \sum_{\substack{l=1 \\ l \neq k}}^p \sum_{\substack{m=1 \\ m \neq l}}^p w_l^c(i) w_m^c(i) (T_l - T_k) \cdot (T_m - T_k) \quad (\text{A.5})$$

$$+ \sum_{\substack{m=1 \\ m \neq k}}^p w_k^c(i) w_m^c(i) \mathbf{0}_e \cdot (T_m - T_k)$$

$$= \sum_{\substack{l=1 \\ l \neq k}}^p \sum_{\substack{m=1 \\ m \neq l}}^p w_l^c(i) w_m^c(i) (T_l - T_k) \cdot (T_m - T_k) \quad (\text{A.6})$$

$$+ w_l^c(i) w_k^c(i) (T_l - T_k) \cdot \mathbf{0}_e$$

$$= \frac{\delta^2}{2} \sum_{\substack{l=1 \\ l \neq k}}^p \sum_{\substack{m=1 \\ m \neq l}}^p w_l^c(i) w_m^c(i) \quad (\text{A.7})$$

$$= \frac{\delta^2}{2} \sum_{\substack{l=1 \\ l \neq k}}^p w_l^c(i) (1 - w_l^c(i) - w_k^c(i)) \quad (\text{A.8})$$

$$= \frac{\delta^2}{2} \left( \sum_{\substack{l=1 \\ l \neq k}}^p w_l^c(i) (1 - w_k^c(i)) - \sum_{\substack{l=1 \\ l \neq k}}^p (w_l^c(i))^2 \right) \quad (\text{A.9})$$

$$= \frac{\delta^2}{2} \left( (1 - w_k^c(i))^2 - \sum_{l=1}^p (w_l^c(i))^2 + (w_k^c(i))^2 \right) \quad (\text{A.10})$$

Finally,

$$\mathcal{A}_{i,k} = \frac{\delta^2}{2} \left( \sum_{l=1}^p (w_l^c(i))^2 - 2w_k^c(i) + 1 \right). \quad (\text{A.11})$$

### A.3 Expression of $Q^\square$

From eqs. (2.74), (2.75) and (2.81), we can define

$$A_i^{\square\top} = C\Omega^s e_i^n, \quad (\text{A.12})$$

$$B_k^{\square\top} = e_k^p, \text{ and} \quad (\text{A.13})$$

$$Q^\square = \sum_{k=1}^p \sum_{\substack{i \\ t_i=T_k}} (A_i^\square - B_k^\square)^\top (A_i^\square - B_k^\square). \quad (\text{A.14})$$

For clarity, the  $\square$  symbol will be dropped temporarily. We have

$$(A_i - B_k)^\top (A_i - B_k) = A_i^\top A_i + B_k^\top B_k - B_k^\top A_i - (B_k^\top A_i)^\top. \quad (\text{A.15})$$

$$\sum_{k=1}^p \sum_{\substack{i \\ t_i=T_k}} A_i^\top A_i = \sum_{i=1}^n A_i^\top A_i \quad (\text{A.16})$$

$$= C\Omega^s \underbrace{\sum_{i=1}^n e_i^n (e_i^n)^\top}_{I_n} \Omega^{s\top} C^\top. \quad (\text{A.17})$$

$$\sum_{k=1}^p \sum_{\substack{i \\ t_i=T_k}} B_k^\top B_k = \sum_{k=1}^p \text{Card}(\{i|t_i = T_k\}) B_k^\top B_k \quad (\text{A.18})$$

$$= \text{Diag}_{k=1}^p (\text{Card}(\{i|t_i = T_k\})) \quad (\text{A.19})$$

$$= CC^\top. \quad (\text{A.20})$$

$$\sum_{k=1}^p \sum_{\substack{i \\ t_i=T_k}} B_k^\top A_i = \sum_{k=1}^p \sum_{\substack{i \\ t_i=T_k}} e_k^p (e_i^n)^\top \Omega^{s\top} C^\top \quad (\text{A.21})$$

$$= \sum_{k=1}^p \begin{bmatrix} \mathbf{0} \\ C[k, :] \\ \mathbf{0} \end{bmatrix} \Omega^{s\top} C^\top \quad (\text{A.22})$$

$$= C\Omega^{s\top} C^\top. \quad (\text{A.23})$$

So finally (mentioning the  $\square$  symbol again),

$$Q^\square = C\Omega^s \Omega^{s\top} C^\top + CC^\top - C\Omega^{s\top} C^\top - C\Omega^s C^\top \quad (\text{A.24})$$

$$= C(\Omega^s \Omega^{s\top} + I_n - \Omega^{s\top} - \Omega^s) C^\top \quad (\text{A.25})$$

$$= C(\Omega^s - I_n)(\Omega^{s\top} - I_n) C^\top. \quad (\text{A.26})$$

#### A.4 Expression of $F$ for equidistant targets

As a reminder (see eq. (2.84) and Appendix A.2),

$$F(T_1, \dots, T_p) = \sum_{k=1}^p \sum_{\substack{i \\ t_i=T_k}} \mathcal{A}_{i,k} \quad (\text{A.27})$$

$$\text{where } \mathcal{A}_{i,k} = \frac{\delta^2}{2} \left( \sum_{l=1}^p (w_l^c(i))^2 - 2w_k^c(i) + 1 \right). \quad (\text{A.28})$$

Then,

$$F(T_1, \dots, T_p) = \frac{\delta^2}{2} \left( \sum_{i=1}^n \sum_{l=1}^p (w_l^c(i))^2 \right. \quad (\text{A.29})$$

$$\left. - 2 \sum_{k=1}^p \sum_{\substack{i \\ t_i=T_k}} w_k^c(i) + \sum_{i=1}^n 1 \right)$$

$$= \frac{\delta^2}{2} \left( |\Omega^c|_{\mathcal{F}}^2 - 2 \sum_{k=1}^p \Omega^c[k, :] C^\top[:, k] + n \right) \quad (\text{A.30})$$

$$= \frac{\delta^2}{2} (|\Omega^c|_{\mathcal{F}}^2 - 2\text{Tr}(\Omega^c C^\top) + n). \quad (\text{A.31})$$

**A.5  $F$  is convex (but not strictly convex)**

Although, the convexity of  $F$  is not used in the development, it is shown here as this is nonetheless related to minimization.

Since  $F$  is a quadratic form in  $U$  defined by a matrix  $Q$  (see Claim 9), it is convex if and only if  $Q$  is positive semi-definite, which can be proved by checking that  $U^\top Q U \geq 0$  for any  $U$ . This is granted by the definition (2.68) of  $F$ .

Note, however, that  $F$  is not strictly convex. Indeed, remembering that  $u_i$  is a weighted sum of the targets with the weights summing to one (see its definition based on eq. (2.10)), it is clear that  $F$  is equal to zero whenever the targets are identical. So there exists  $U$  not equal to  $\mathbf{0}_{pe}$  such that  $U^\top Q U = 0$ .

### A.6 Infinite $\gamma$ with the Inverse Function as Weight

Let  $x$  be a sample not in the learning set. If the weighting function is defined as follows

$$\forall x_i, i \in [1..n], w^{\text{rad}}(\gamma|x - x_i|) = \frac{1}{1 + \gamma|x - x_i|}, \quad (\text{A.32})$$

then

$$(2.14) \Leftrightarrow w_j^s(x) = \left( \sum_{i=1}^n \frac{1 + \gamma|x - x_j|}{1 + \gamma|x - x_i|} \right)^{-1}. \quad (\text{A.33})$$

Therefore,

$$\lim_{\gamma \rightarrow +\infty} w_j^s(x) = \left( \sum_{i=1}^n \frac{|x - x_j|}{|x - x_i|} \right)^{-1}. \quad (\text{A.34})$$

Let us make two remarks. First, we have

$$\lim_{\gamma \rightarrow +\infty} \frac{w_i^s(x)}{w_j^s(x)} = \frac{|x - x_j|}{|x - x_i|}, \quad (\text{A.35})$$

so the ratio between the weights associated to two learning samples simply tends toward a distance-to-learning-sample ratio.

Second, far away from the learning samples, all the distances from  $x$  to the learning samples tend to be equal. So the weight  $w_j^s(x)$  tends to  $1/n$ . Hence, the samples away from the learning set are transformed into a common point, so that the proposed classifier is unusable.



Appendix B

**List of handcrafted features**



**Feature from Zooprocess**

<b>Name</b>	<b>Description</b>
area	Surface area of the object in square pixels
meanpos	Average grey value within the object; sum of the grey values of all pixels in the object divided by the number of pixels
stddev	Standard deviation of the grey value used to generate the mean grey value
mode	Modal grey value within the object
Minor	Minimum grey value within the object (0 = black)
max	Maximum grey value within the object (255 = white)
x	X position of the center of gravity of the object in the smallest rectangle enclosing the object
y	Y position of the center of gravity of the object in the smallest rectangle enclosing the object
xstart	X coordinate of the top left point of the image in the smallest rectangle enclosing the object
ystart	Y coordinate of the top left point of the image in the smallest rectangle enclosing the object
XMg5	X position of the center of gravity of the object's grey level in the smallest rectangle enclosing the object
YMg5	Y position of the center of gravity of the object's grey level in the smallest rectangle enclosing the object
xmg5	X position of the center of gravity of the object, using a gamma value of 5
ymg5	Y position of the center of gravity of the object, using a gamma value of 5
bx	X coordinate of the top left point of the smallest rectangle enclosing the object
by	Y coordinate of the top left point of the smallest rectangle enclosing the object
width	Width of the smallest rectangle enclosing the object
height	Height of the smallest rectangle enclosing the object
perim	The length of the outside boundary of the object
major	Primary axis of the best fitting ellipse for the object
minor	Secondary axis of the best fitting ellipse for the object
angle	Angle between the primary axis and a line parallel to the x-axis of the image
circ	Circularity = $(4 * \text{Pi} * \text{Area}) / \text{Perim}^2$ ; a value of 1 indicates a perfect circle, a value approaching 0 indicates an increasingly elongated polygon
feret	Maximum feret diameter, i.e., the longest distance between any two points along the object boundary
intden	Integrated density. This is the sum of the grey values of the pixels in the object (i.e. = Area*Mean)
median	Median grey value within the object
skew	Skewness of the histogram of grey level values
kurt	Kurtosis of the histogram of grey level values
%area	Zooscan, FlowCam and Generic : Percentage of object's surface area that is comprised of holes, defined as the background grey level UVP5 and UVP6 : 1 - Percentage of object's surface area that is comprised of holes, defined as the background grey level
area_exc	Zooscan, FlowCam and Generic : Surface area of the object excluding holes, in square pixels (=Area*(1-(%area/100)) UVP5 and UVP6 : Surface area of the holes in the object, in square pixels (=Area*(1-(%area/100)))
fractal	Fractal dimension of object boundary (Berube and Jebrak 1999)
skelarea	Surface area of skeleton in pixels. In a binary image, skeleton is obtained by repeatedly removing pixels from the edges of objects until they are reduced to the width of a single pixel.
slope	Slope of the grey level normalized cumulative histogram
histcum1	Grey level value at the first quartile of the normalized cumulative histogram of grey levels
histcum2	Grey level value at the second quartile of the normalized cumulative histogram of grey levels
histcum3	Grey level value at the third quartile of the normalized cumulative histogram of grey levels
nb1	Number of remaining objects in the image after thresholding on level Histcum1
nb2	Number of remaining objects in the image after thresholding on level Histcum2
nb3	Number of remaining objects in the image after thresholding on level Histcum3
symetrieH	Bilateral horizontal symmetry index
symetrieV	Bilateral vertical symmetry index
symetrieHc	Symmetry of the largest remaining object in relation to the horizontal axis after thresholding at the grey level Histcum1 value
symetrieVc	Symmetry of the largest remaining object in relation to the vertical axis after thresholding at the grey level Histcum1 value
convperim	The perimeter of the smallest polygon within which all points in the object fit
convarea	The area of the smallest polygon within which all points in the object fit
fcons	Measure of contrast based in the texture feature descriptor (Amadasun and King, 1989)
thickr	Thickness ratio : relation between the maximum thickness of an object and the averag thickness of the object excluding the maximum
tag	no more utilized
esd	Equivalent Spherical Diameter
elongation	major / minor
range	max - min
meanpos	$(\text{mean-max}) / (\text{mean-min})$
centroids	$\sqrt{(\text{pow}(\text{xm-x},2) + \text{pow}(\text{ym-y},2))}$
cv	$100 * (\text{stddev}/\text{mean})$
sr	$100 * (\text{stddev}/(\text{max-min}))$
perimareaexc	$\text{perim}/(\sqrt{(\text{area\_exc})})$
feretareaexc	$\text{feret}/(\sqrt{(\text{area\_exc})})$
perimmajor	perim/major
perimferet	perim/feret
circex	$(4 * \text{PI} * \text{area\_exc}) / (\text{pow}(\text{perim},2))$
cdexc	$(1/(\sqrt{(\text{area\_exc})})) * \sqrt{(\text{pow}(\text{xm-x},2) + \text{pow}(\text{ym-y},2))}$

**Figure B.1** List of all the Zooprocess features and their definition, original file from <https://sites.google.com/view/piqv/softwares/flowcamzooscan>

Feature name	Zooscan	UVP5-HD
area	×	×
mean	×	×
stddev	×	×
mode	×	×
min	×	×
max	×	×
perim.	×	×
width	×	×
height	×	×
major	×	×
minor	×	×
circ.	×	×
feret	×	×
intden	×	×
median	×	×
skew	×	×
kurt	×	×
% area	×	×
exc	×	×
fractal	×	×
skelarea	×	×
slope	×	×
histcum1	×	×
histcum2	×	×
histcum3	×	×
nb1	×	×
nb2	×	×
symetrieH	×	×
symetrieV	×	×
symetrieHc	×	×
symetrieVc	×	×
convperim	×	×
convarea	×	×
fcons	×	×
thickr	×	×
esd	×	×
elongation	×	×
range	×	×
centroids	×	×
sr	×	×
perimareaexc	×	
feretareaexc	×	
perimferet	×	×
perimmajor	×	×
circex	×	×
cdexc	×	

Table B.1 List of the Zooprocess features used for the experiemnts of chapter 4.



## Appendix C

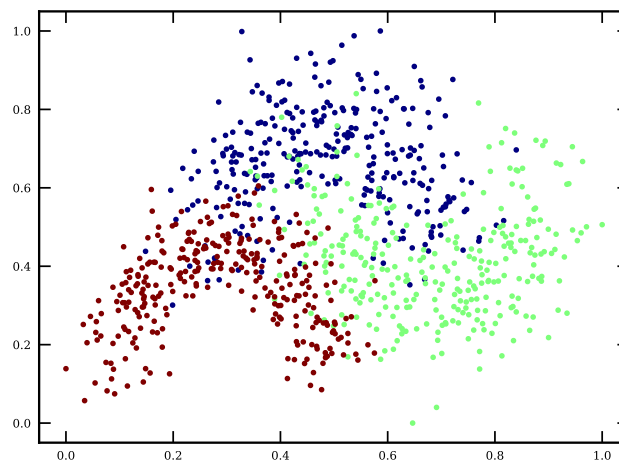
### Additional figures

The numerical version of this appendix document is recommended for accurate reading of the figures.

#### C.1 Experiment: Influence of targets' definition on classification

This section gives details and additional figures about the experiment on the influence of the targets' definition on the classification, see section 2.5.3.

The  $n = 900$  learning samples used for the experiment are shown on fig. C.1. The  $m = 900$  test samples used to compute the empirical risk are drawn from the same distributions. Let us remind that the transformation is computed for fixed parameter  $\gamma = 100$  ( $\gamma_i = \gamma \forall i \in [1..n]$ ). On the other hand, for each iteration of the experiment, among 1000, the targets are drawn from a uniform distribution (in the range  $[0, 1] \times [0, 1]$ ). The results of the experiment, *i.e.*, the empirical risk as a function of the absolute difference between the extrema of the target triangle angles ( $\phi, \theta, \psi$ , in degrees) with the empirical risk, are given on fig. C.2. As mentioned in the main text (section 2.5.3), we observe that the lowest values of the empirical risk are obtained for low differences between the extreme angles, *i.e.*, for equidistant targets and configurations close to it. This is in accordance with the Conjecture 2.2. Figures C.3 and C.4 present configurations of targets for small and large values of empirical risk.



**Figure C.1** Learning samples, each colour represents a class. The 'moon distributions' were used.

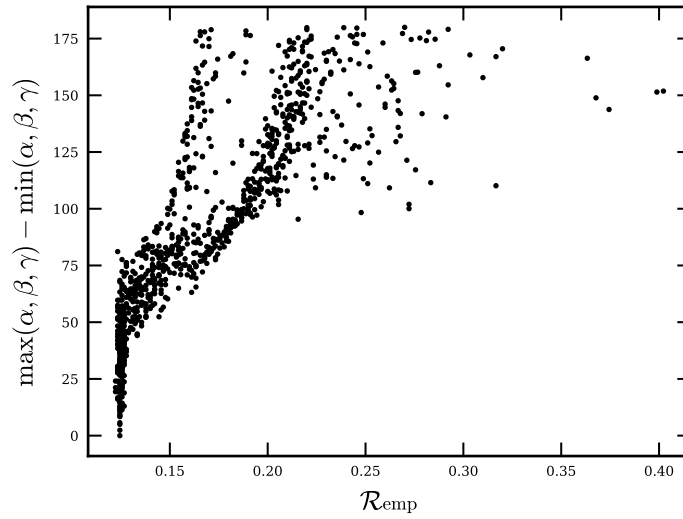


Figure C.2 Result of the experiment. Same as fig. 2.3.

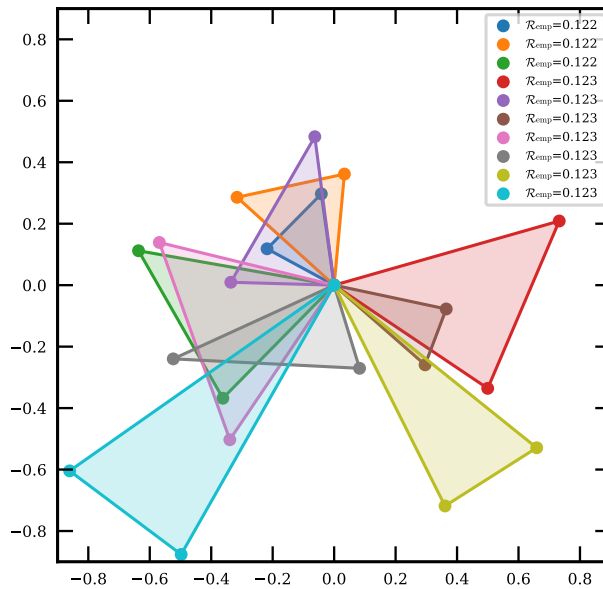
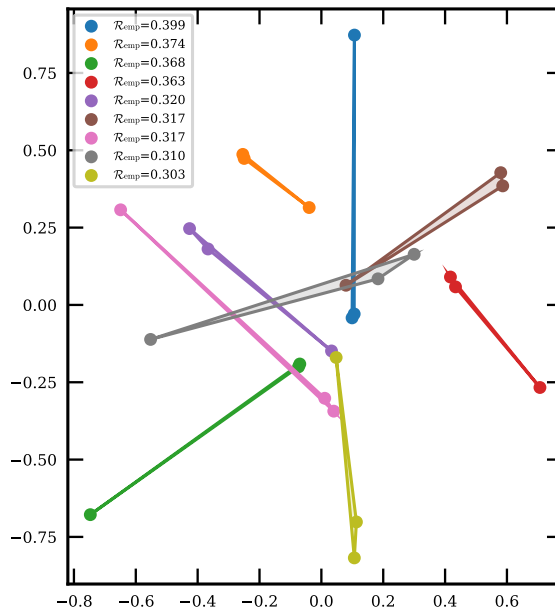
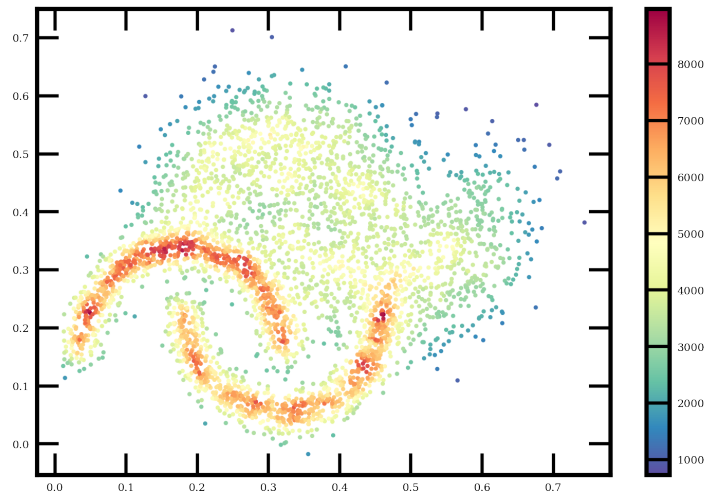


Figure C.3 Representation of the ten target sets (dots) that returned small empirical risks over the test set (better classification). Each colour represent a set of targets. Lines between targets have been added, and each target set has been translated to improve the reading of the figure.



**Figure C.4** Representation of the ten target sets (dots) that returned large empirical risks over the test set (worst classification). Each colour represent a set of targets. Lines between targets have been added to improve the reading of the figure.

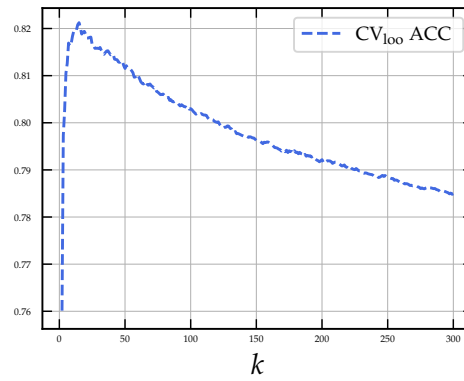
**C.2 Parameter  $\gamma_i$  with t-SNE**



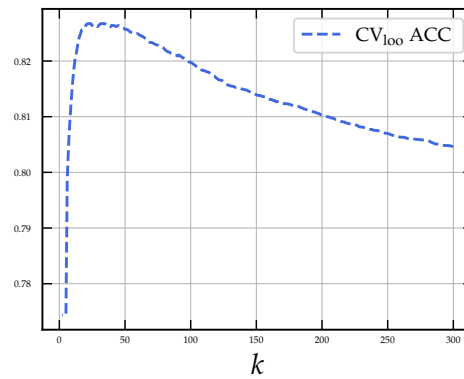
**Figure C.5** Each dot is a sample, the colour gives the value of  $\gamma_i, i \in [1 \dots n]$ , computed with the t-SNE method [Van der Maaten and Hinton, 2008].

### C.3 CIFAR-10

#### C.3.1 $d=100$ features

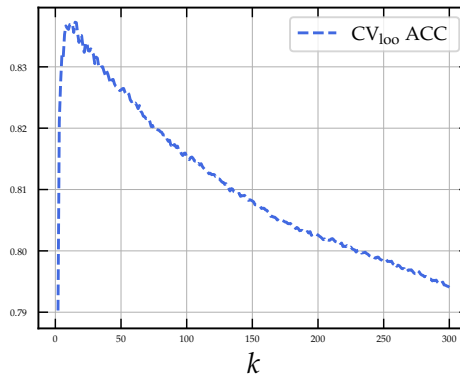


**Figure C.6**  $k$ -NN classification, for  $d = 100$ . Evolution of the ACC with respect to the number of neighbours  $k$ , all predictions from 1 to 300.

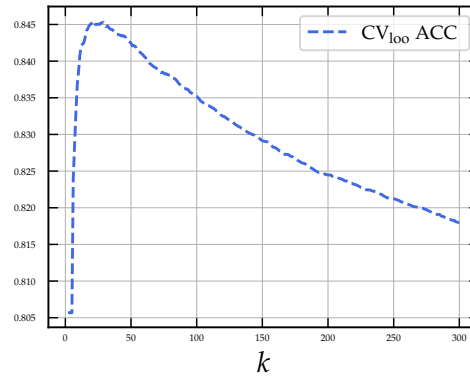


**Figure C.7**  $W$ - $k$ -NN classification, for  $d = 100$ . Evolution of the ACC with respect to the number of neighbours  $k$ , all predictions from 1 to 300.

#### C.3.2 All features ( $d=1000$ )



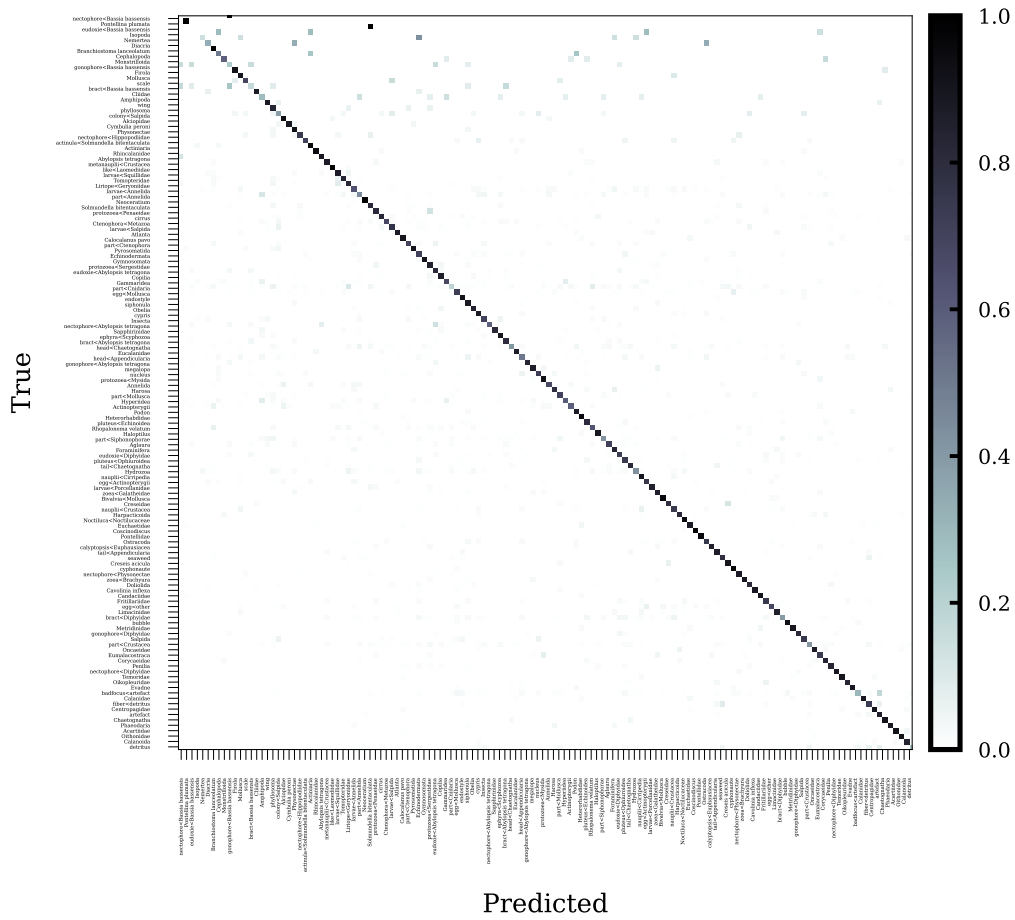
**Figure C.8**  $k$ -NN classification, for  $d = 1000$  (all extracted features). Evolution of the ACC with respect to the number of neighbours  $k$ , all predictions from 1 to 300.



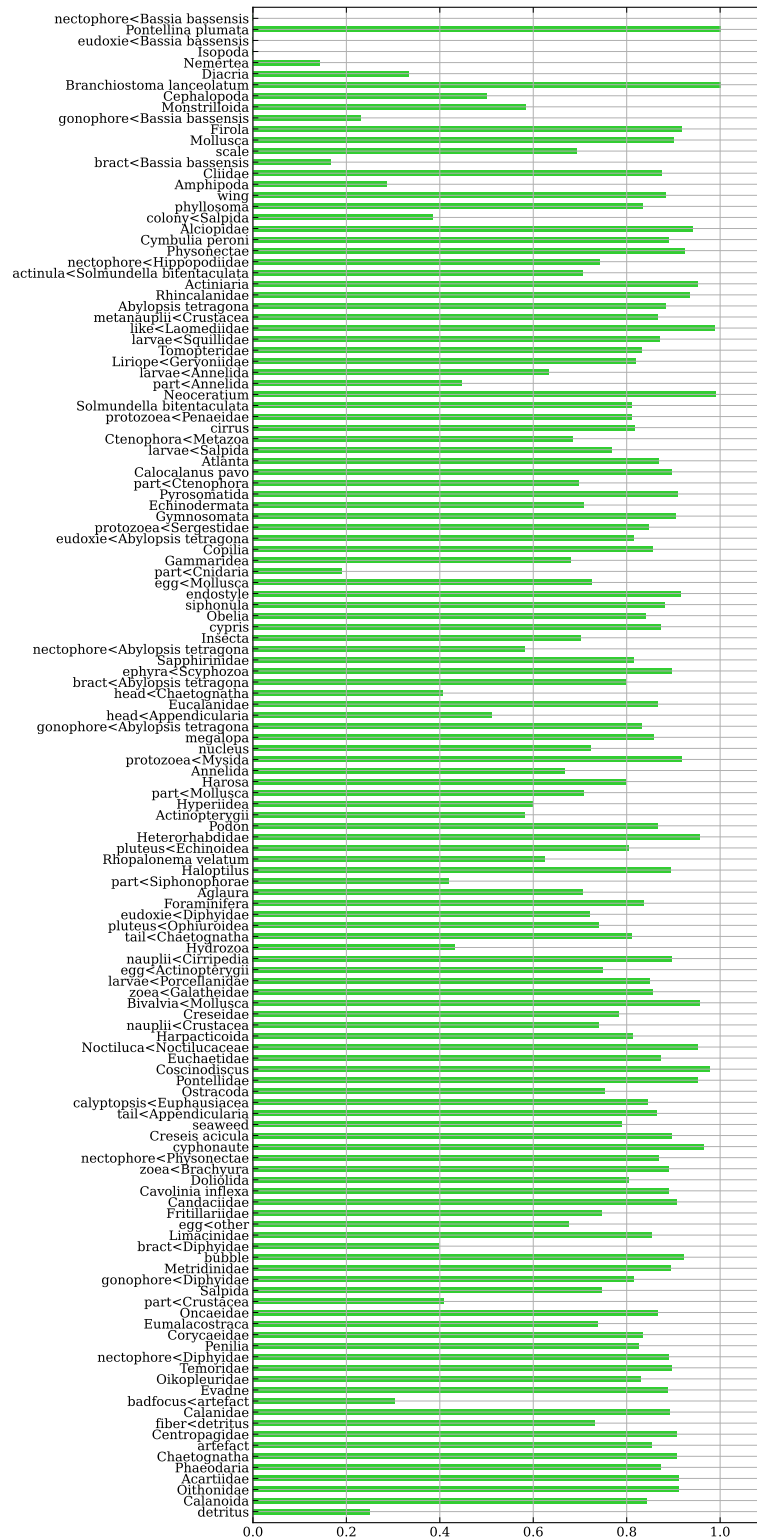
**Figure C.9**  $W$ - $k$ -NN classification, for  $d = 1000$  (all extracted features). Evolution of the ACC with respect to the number of neighbours  $k$ , all predictions from 1 to 300.

## C.4 Zooscan data set



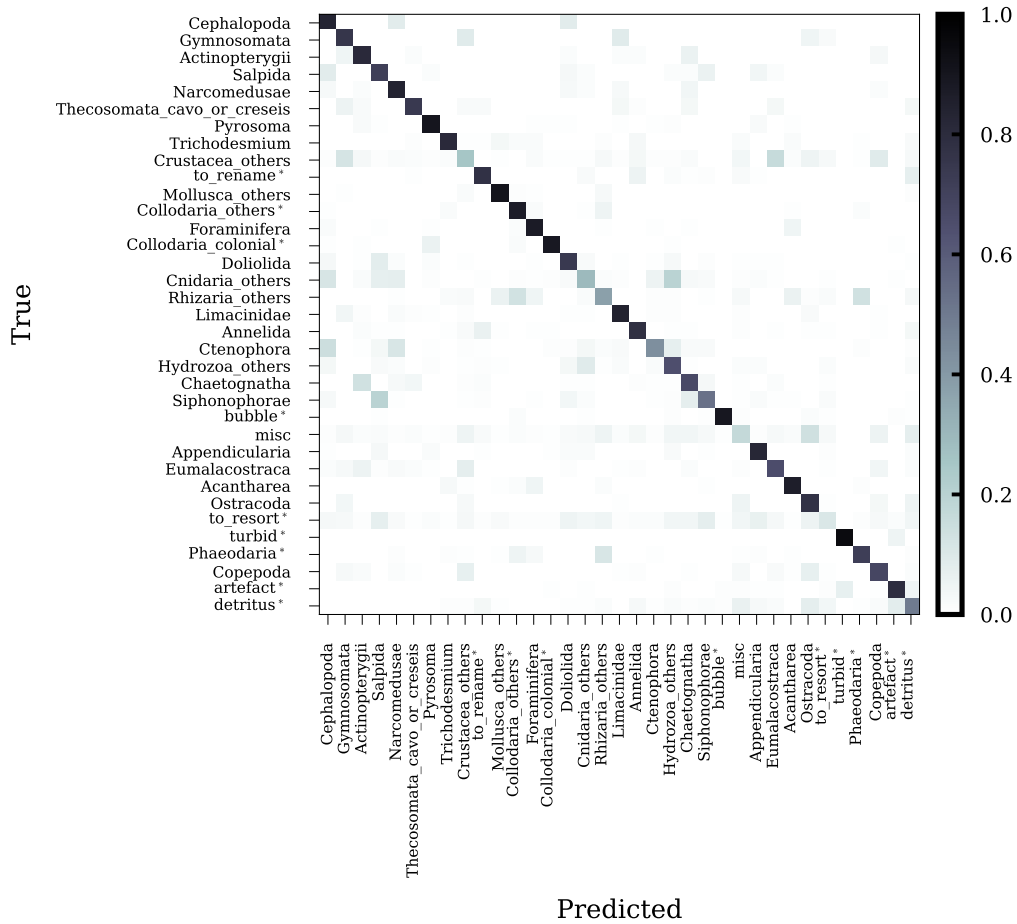


**Figure C.10** Normalized (by the number of element per class) confusion matrix of the  $W$ - $k$ -NN predictions on the test set, based on the *deep*-features extracted from a fine-tuned CNN ( $d = 10$ , see details in section 4.2.3). The number of neighbours is  $k^* = 110$ . The classes are organized from the less represented to the more represented (see fig. 4.2). Note the proportion of *detritus* correctly classified (the last pixel on the bottom right) is low.

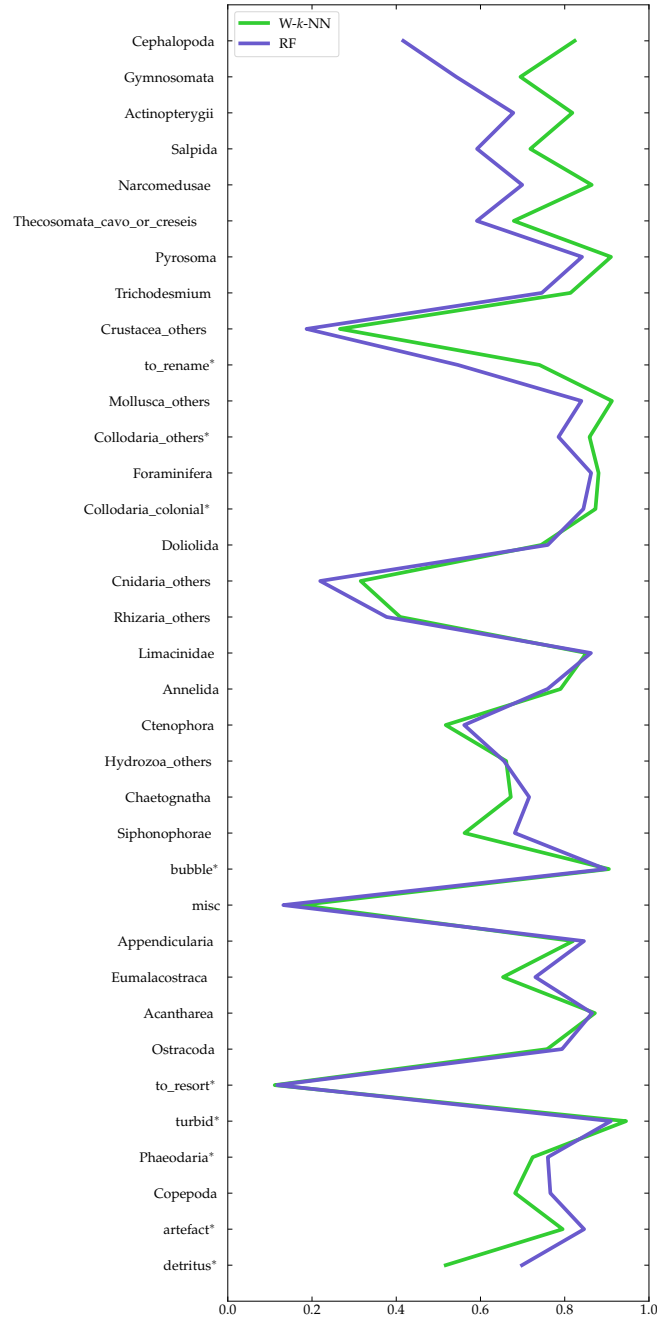


**Figure C.11** Recall (number of samples correctly classified in a class, relative to the number of elements in that class) of the  $W$ - $k$ -NN predictions on the ZooScanNet test set, based on the *deep*-features extracted from a fine-tuned CNN ( $d = 10$ , see details in section 4.2.3). The number of neighbours is  $k^* = 110$ . The classes are organized from the less represented to the more represented (see fig. 4.2). Note the proportion of *detritus* correctly classified is low.

C.5 UVP5-HD data set



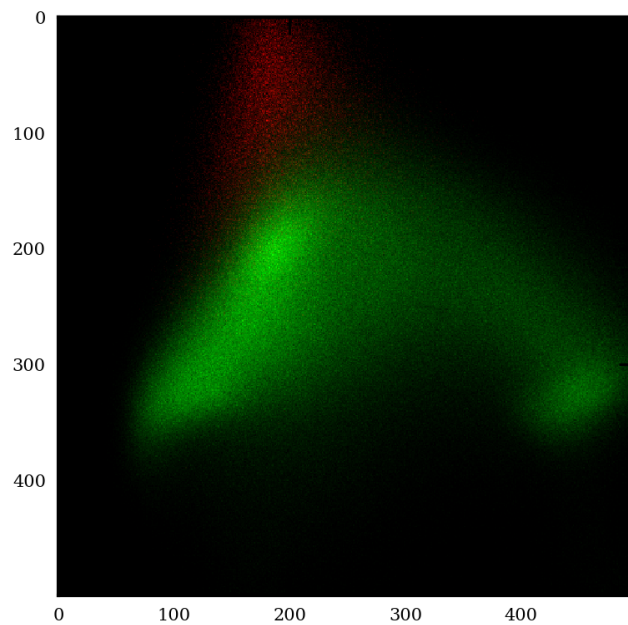
**Figure C.12** Normalized (by the number of element per class) confusion matrix of the  $W$ - $k$ -NN predictions on the test set, based on the *deep*-features extracted from a fine-tuned CNN, for the UVP5-HD data set ( $d = 10$ , see details in section 4.3.2). The number of neighbours is  $k^* = 580$ . The classes are organized from the less represented to the more represented (see fig. 4.5).



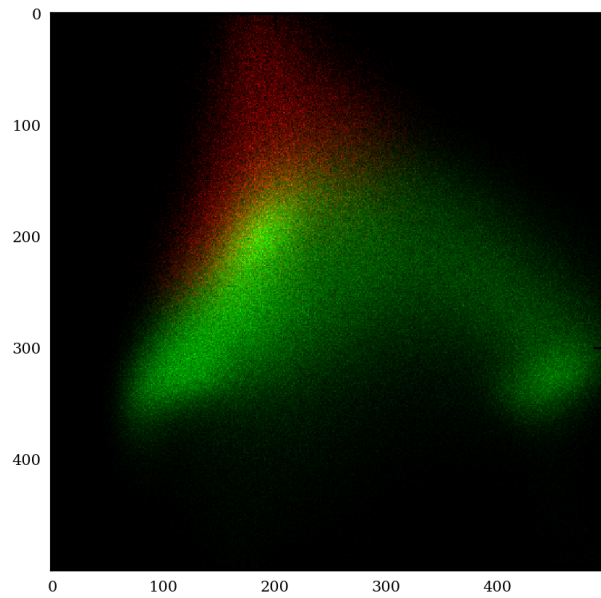
**Figure C.13** Recall score (number of samples correctly classified in a class, relative to the number of elements in that class) for the classification of the test set based on the first 10 components of the *deep*-features with the *W-k-NN* ( $k^* = 580$ ) for the UVP5-HD data set.

### C.6 UVP5-HD Copepods

Figure C.14 is a 2-d histogram of 2 dimensions of the samples (among 10, see details in sections 4.2.3 and 4.3.3) of the UVP5-HD data set. The final image is a superposition of two 2-d histograms. One in green for the non-copepod samples, and another one in red for the copepod samples. For both, the colour represents the true relative density (*i.e.*, density for each class), black indicates zero density. Figure C.15 is the same, but the two colours are for the predicted labels, as opposed to the true labels for fig. C.14. With both figures, we can conjecture that the classifier tends to over-predict non-copepod samples as copepod, which is in accordance with the high values of recall and low values of precision obtained for the copepods (see section 4.3.3). Note that the same figures with the absolute density would be more useful to conclude, but, due to the high class imbalance, the copepods (red) would not be visible.



**Figure C.14** 2d-histogram of 2 dimensions of the samples. Colour intensity represent the relative density ; red for copepod samples and green for non-copepod samples.



**Figure C.15** 2d-histogram of 2 dimensions of the samples. Colour intensity represent the relative density ; red for samples predicted as copepod and green for samples predicted as non-copepod.



## Appendix D

### Projection of an ellipsoid: Development details

This appendix aims at giving details for the reading of chapter 7.

#### D.1 Axes-align ellipsoid

An ellipsoid centred on the origin is composed of the ensemble of 3-d points  $x$  verifying

$$x^T M x = 1 \quad (\text{D.1})$$

where  $M$  is a real, symmetric, positive definite,  $3 \times 3$ -matrix whose elements are denoted by  $m_{ij}$ . The volume of the ellipsoid is defined by

$$V = \frac{4}{3} \frac{\pi}{\sqrt{\det(M)}}. \quad (\text{D.2})$$

Matrix  $M$  encodes the overall size, shape (semi-axes ratios) and orientation of the ellipsoid. It can be written using a block matrix notation

$$M = \left[ \begin{array}{c|c} M_{11} & M_{21}^T \\ \hline M_{21} & m_{33} \end{array} \right] \quad (\text{D.3})$$

where  $m_{33}$  is a scalar (the dimensions of the other terms follow). If the ellipsoid is aligned on the axes of the coordinate system, then its form is

$$M = \begin{bmatrix} 1/r_1^2 & 0 & 0 \\ 0 & 1/r_2^2 & 0 \\ 0 & 0 & 1/r_3^2 \end{bmatrix} \quad (\text{D.4})$$

where the  $r_i$ 's are the semi-axes.

#### D.2 Deriving $S_e$

To propose a more explicit form of  $S_e$ , let us use the following block matrix formulation

$$M = \left[ \begin{array}{c|c} M_{11} & M_{21}^T \\ \hline M_{21} & m_{33} \end{array} \right] \quad (\text{D.5})$$

where  $M_{11}$  is a  $2 \times 2$ -matrix,  $M_{21}$  is a  $1 \times 2$ -vector, and  $m_{33}$  is a scalar. Using such a block formulation, we have

$$e e^T = \left[ \begin{array}{c|c} 0_{11} & 0_{21}^T \\ \hline 0_{21} & \epsilon^2 \end{array} \right] \quad (\text{D.6})$$



where  $0_{ij}$  denotes a matrix of zeros matching the dimension of  $M_{ij}$ .  
Then

$$Me e^T M = \epsilon^2 \left[ \begin{array}{c|c} M_{21}^T M_{21} & m_{33} M_{21}^T \\ \hline m_{33} M_{21} & m_{33}^2 \end{array} \right]. \quad (\text{D.7})$$

Similarly

$$e^T Me = \epsilon^2 m_{33}. \quad (\text{D.8})$$

Therefore

$$S_\epsilon = \left[ \begin{array}{c|c} M_{21}^T M_{21} & m_{33} M_{21}^T \\ \hline m_{33} M_{21} & m_{33}^2 \end{array} \right] - \left( m_{33} - \frac{1}{\epsilon^2} \right) M \quad (\text{D.9})$$

$$= \left[ \begin{array}{c|c} M_{21}^T M_{21} - m'_{33} M_{11} & (m_{33} - m'_{33}) M_{21}^T \\ \hline (m_{33} - m'_{33}) M_{21} & (m_{33} - m'_{33}) m_{33} \end{array} \right] \quad (\text{D.10})$$

where  $m'_{33}$  is defined as

$$m'_{33} = m_{33} - \frac{1}{\epsilon^2}. \quad (\text{D.11})$$

So finally

$$S_\epsilon = \left[ \begin{array}{c|c} M_{21}^T M_{21} - m'_{33} M_{11} & (1/\epsilon^2) M_{21}^T \\ \hline (1/\epsilon^2) M_{21} & (1/\epsilon^2) m_{33} \end{array} \right]. \quad (\text{D.12})$$

### D.3 Semi-axes for perspective projection

Let  $\lambda_1$  and  $\lambda_2$  be the two (positive) eigenvalues of  $P$ ,  $\lambda_1 \leq \lambda_2$ . Then the semi-minor and semi-major axes of the ellipse defined by eq. (7.21) are

$$\rho_i = \sqrt{\frac{r - Qc/2}{\lambda_i}}, i \in \{1, 2\}. \quad (\text{D.13})$$

Gathering everything together,  $\rho_i$  can be rewritten in terms of  $M$  as follows

$$r = -(1 - \delta/\epsilon)^2 m_{33}, \quad (\text{D.14})$$

$$Q = 2 \frac{\epsilon - \delta}{\epsilon^2} M_{21}, \quad (\text{D.15})$$

$$P = M_{21}^T M_{21} - \left( m_{33} - \frac{1}{\epsilon^2} \right) M_{11}, \quad (\text{D.16})$$

$$c = -P^{-1} Q^T / 2, \quad (\text{D.17})$$

$$\lambda_i = (\text{tr}(P) + \sigma_i \sqrt{\Delta}) / 2, \quad (\text{D.18})$$

$$|\sigma_i| = 1 \text{ and } \sigma_1 \sigma_2 = -1, \quad (\text{D.19})$$

$$\Delta = \text{tr}(P)^2 - 4 \det(P) \quad (\text{D.20})$$

where  $\text{tr}(P)$  is the trace of  $P$ ,  $\det(P)$  is its determinant, and the  $\sigma_i$ 's are chosen so that  $\rho_1 \geq \rho_2$ .

For a parallel projection (*i.e.*,  $\epsilon = \infty$ ), the semi-minor and semi-major axes have the following simpler expression

$$\rho_i = \sqrt{\frac{m_{33}}{\lambda_i}}, i \in \{1, 2\} \quad (\text{D.21})$$

with

$$P = m_{33} M_{11} - M_{21}^T M_{21}, \quad (\text{D.22})$$

while  $\lambda_i$ ,  $\sigma_i$ , and  $\Delta$  are unchanged. Note that  $\delta$  no longer appears in the equations.

## Appendix E

### The proposed method, step-by-step

This section gathers the results of the different sections into a step-by-step procedure for estimating the total volume of copepods given a data set of 2-D views. It is composed of two stages: a learning stage which has to be performed once for all, or whenever the expert thinks the proposed simulation procedure must be adapted to the data, and a “usage” stage which can be applied at will.

#### E.1 Learning stage

1. Generate random ellipsoid samples that realistically represent a generic population of copepods, or a population following some characteristics inferred from the data set. The randomness must be constrained by the expert knowledge in the form of specific simulation parameters.
2. Compute the total volume of the ellipsoid samples. This represents the true total volume. See eqs. (8.1) and (D.2).
3. For each ellipsoid sample, compute the projection ellipse (see eq. (7.22)) and the estimated volume using either the  $\mathcal{M}_{\text{ESD}}$  (see eq. (5.1)) or the  $\mathcal{M}_{\text{ELL}}$  method (see eq. (5.2)).
4. Sum all the estimated volumes to get the estimated total volume.
5. Compute the total volume estimation error  $\mathcal{T}_*$  from the true and estimated total volumes (see eq. (8.1)). This is the final product of the learning stage.

#### E.2 ‘Prediction’ stage

1. For each copepod image of a data set, determine the copepod silhouette using an image segmentation method. On UVP images, a simple binarization using a fixed threshold is enough.
  - 1.a. For the  $\mathcal{M}_{\text{ESD}}$  method, compute the silhouette area  $A$  (see section 6.2.2) and the corresponding estimated volume (see eq. (5.1)).
  - 1.b. For the  $\mathcal{M}_{\text{ELL}}$  method, fit an ellipse onto the silhouette (see section 6.2.2). Let  $\rho_1$  and  $\rho_2$  be the semi-major and semi-minor axes, respectively. Then compute the corresponding estimated volume (see eq. (5.2)).
2. Sum all the estimated volumes to get the estimated total volume  $\tilde{W}_*$  where  $*$  is either ESD or ELL.

3. Compute the corrected total volume estimation  $\hat{W}_*$  by dividing  $\tilde{W}_*$  with  $\mathcal{T}_*$  from the learning stage (see eq. (8.2)).

### E.3 Uniformly random rotations

This section defines the rotation matrices used to simulate random orientations of ellipsoids.

In order to generate an ellipsoid with a uniformly random orientation, we generate a random rotation matrix  $R$  and rotate an axis-aligned ellipsoid with it. The generation of an axis-aligned ellipsoid is described in section 7.4. If the axis-aligned ellipsoid is represented by a matrix  $M$  (see eq. (D.3)), then the rotated ellipsoid is represented by the matrix

$$M_{\text{rot}} = RMR^\top. \quad (\text{E.1})$$

A general rotation matrix can be defined using three elementary rotation matrices

$$R = R_z(\Phi)R_y(\Theta)R_x(\Psi) \quad (\text{E.2})$$

with  $R_i(\alpha)$ ,  $i \in \{x, y, z\}$ , defines the rotation by angle  $\alpha$  around axis  $i$ . To generate a random rotation matrix, one has to randomly choose the angle triplet  $(\Psi, \Theta, \Phi)$ . In order to guarantee the uniformity of the ellipsoid orientations, the angles  $\Psi$ ,  $\Theta$ , and  $\Phi$  must be distributed adequately, that is

$$\Psi = U[0, 2\pi[ \quad (\text{E.3})$$

$$\Theta = \arccos(1 - 2U[0, 1]) - \frac{\pi}{2} \quad (\text{E.4})$$

$$\Phi = U[0, 2\pi[ \quad (\text{E.5})$$

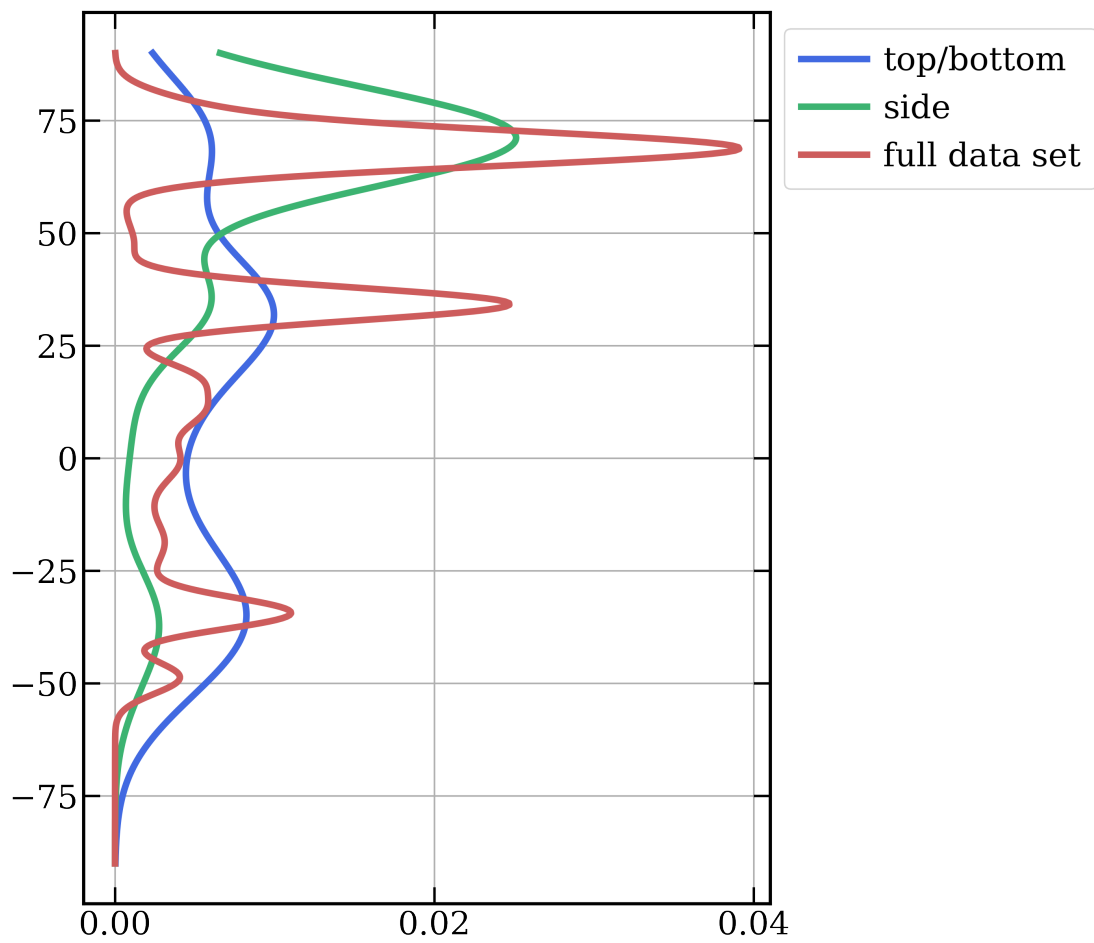
where  $U[a, b[$  is the uniform distribution between  $a$  (included) and  $b$  (excluded).

## Appendix F

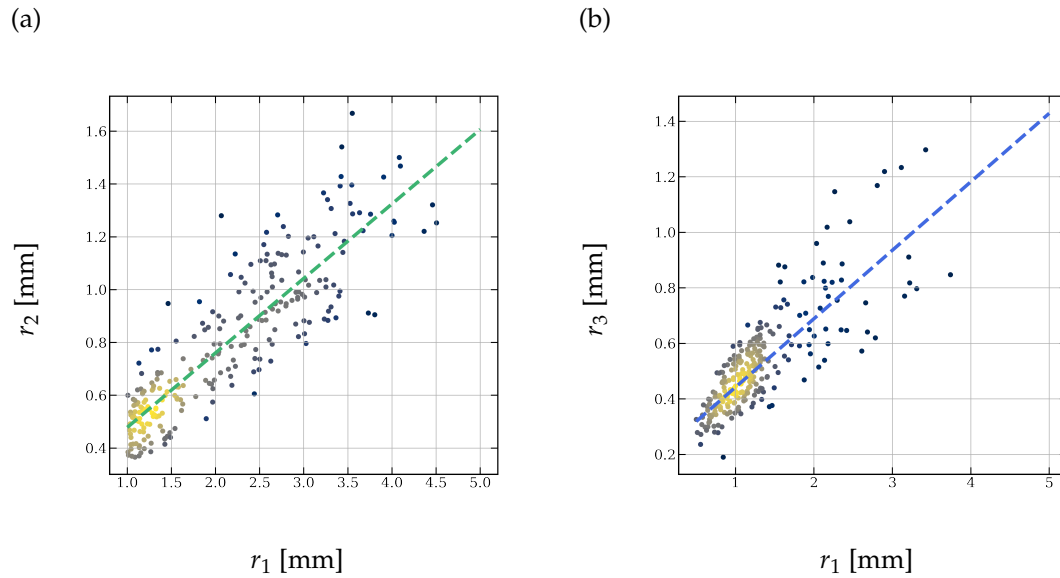
### Distribution of selected sample images

To define the real-world distribution of the semi axes of the ellipsoid representing the body of copepods ( $r_1$ ,  $r_2$ , and  $r_3$ ) as well as the ratios between them, defining the shape of the ellipsoid ( $r_2/r_1$  and  $r_3/r_1$ ), 295 copepods seen from the side (on which  $r_1$  and  $r_2$  are measurable) and 265 copepods seen from the top or bottom (on which  $r_1$  and  $r_3$  are measurable) were manually curated from a collection of >150k images. To make sure that these small samples were representative of the whole data set, we checked their latitudinal and size (i.e.  $r_1$ ) distributions. The shape of the latitudinal distribution of the side and top/bottom views matches well that of the total data set (fig. F.1). The side views show an excess at high latitude, likely linked with a bias in the size distribution (see below; copepods are larger at high latitudes), and a linked under-representation elsewhere. The pattern is opposite for the top-bottom views. However, no region is completely missed in the samples and even some details of the distribution (such as the two peaks around  $-40^\circ$ ) are captured. Therefore, we consider them representative enough.

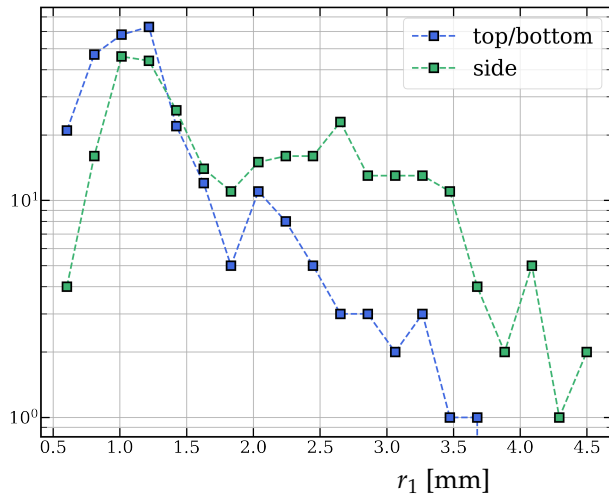
The length distribution is expected to be an exponential decay [Sprules and Barth, 2016], *i.e.*, a linear decrease, in log-scale. This is approximately true once the lower detection limit of the camera is passed, after  $\sim 1$  mm (fig. F.3). However, the distribution of side views shows an excess in the size range 2 to 3.5 mm. This is likely due to the fact that telling that a copepod is viewed from the top/bottom can be determined from the geometry of its antennas relative to its body, no matter its size; making sure that a copepod is viewed from the side requires additional details, which are easier to assess on larger individuals, inducing a bias in the manual selection of images. As explained in the main text, this has little consequence on the estimation of the distribution of the semi-axes ratios ( $r_2/r_1$  and  $r_3/r_1$ ) but does not allow the estimation of the distribution of  $r_1$  from these samples only.



**Figure F.1** Kernel density estimate of the latitudinal distribution of the images of all copepods and of the side or top/bottom views.



**Figure F.2** (a) Relationship between  $r_2$  and  $r_1$  from 254 copepods seen from the side, for  $r_1 > 1$  mm. (b) Relationship between  $r_3$  and  $r_1$  from the 173 copepods seen from the top or bottom, for  $r_1 > 1$  mm. The coloured dashed line are linear regressions fits, significant in both cases ( $p < 0.01$ ,  $R^2 = 88\%$  for (a) and  $R^2 = 75\%$  for (b)). The colour scale of points represents the density of samples.



**Figure F.3** Distribution of the length of the semi-major axis of the ellipse fitted in the two views of the copepods. The vertical axis is the number of observations, in  $\log_{10}$  scale. The horizontal axis is the semi-major axis  $r_1$ , which is equal to  $\rho_1$  in these viewpoints and approximates the half of the prosome length, in millimeters.



## Bibliography

- M. J. Behrenfeld, J. T. Randerson, C. R. McClain, et al. Biospheric primary production during an ENSO transition. *Science*, 291(5513):2594–2597, 2001.
- M. Benfield, C. Davis, and S. Gallager. Estimating the in-situ orientation of calanus finmarchicus on georges bank using the video plankton recorder. *Plankton Biology and Ecology*, 47:69–72, 01 2000.
- M. C. Benfield, C. S. Davis, P. H. Wiebe, S. M. Gallager, R. Gregory Lough, and N. J. Copley. Video plankton recorder estimates of copepod, pteropod and larvacean distributions from a stratified region of georges bank with comparative measurements from a moose sampler. *Deep Sea Research Part II: Topical Studies in Oceanography*, 43(7):1925–1945, 1996. ISSN 0967-0645. doi:[https://doi.org/10.1016/S0967-0645\(96\)00044-6](https://doi.org/10.1016/S0967-0645(96)00044-6). URL <https://www.sciencedirect.com/science/article/pii/S0967064596000446>.
- M. C. Benfield, P. Grosjean, P. F. Culverhouse, X. Irigoien, M. E. Sieracki, A. Lopez-Urrutia, H. G. Dam, Q. Hu, C. S. Davis, A. Hansen, et al. Rapid: research on automated plankton identification. *Oceanography*, 20(2):172–187, 2007.
- J. L. Bentley. Multidimensional binary search trees used for associative searching. *Commun. ACM*, 18(9):509–517, 1975.
- L. Breiman. Random forests. *Machine learning*, 45(1):5–32, 2001.
- E. Buitenhuis, C. Le Quéré, O. Aumont, G. Beaugrand, A. Bunker, A. Hirst, T. Ikeda, T. O’Brien, S. Piontkovski, and D. Straile. Biogeochemical fluxes through mesozooplankton. *Global Biogeochemical Cycles*, 20(2), 2006.
- E. T. Buitenhuis, M. Vogt, R. Moriarty, N. Bednaršek, S. C. Doney, K. Leblanc, C. Le Quéré, Y.-W. Luo, C. O’Brien, T. O’Brien, J. Peloquin, R. Schiebel, and C. Swan. MAREDAT: towards a world atlas of MARine Ecosystem DATA. *Earth System Science Data*, 5(2):227–239, 2013.
- B. Charlier, J. Feydy, J. A. Glaunès, F.-D. Collin, and G. Durif. Kernel operations on the gpu, with autodiff, without memory overflows. *Journal of Machine Learning Research*, 22(74):1–6, 2021. URL <http://jmlr.org/papers/v22/20-275.html>.
- T. Chen and C. Guestrin. Xgboost: A scalable tree boosting system. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, KDD ’16*, page 785–794, New York, NY, USA, 2016. Association for Computing Machinery. ISBN 9781450342322. doi:[10.1145/2939672.2939785](https://doi.org/10.1145/2939672.2939785). URL <https://doi.org/10.1145/2939672.2939785>.



- L. Chizat and F. Bach. On the global convergence of gradient descent for over-parameterized models using optimal transport. In S. Bengio, H. Wallach, H. Larochelle, K. Grauman, N. Cesa-Bianchi, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 31. Curran Associates, Inc., 2018. URL <https://proceedings.neurips.cc/paper/2018/file/a1afc58c6ca9540d057299ec3016d726-Paper.pdf>.
- D. Conway. Identification of the copepodite developmental stages of twenty-six north atlantic copepods. occasional publication of the marine biological association no. 21 (revised edition). Technical report, Marine Biological Association of the United Kingdom, Plymouth (UK), 2012. URL <http://plymsea.ac.uk/id/eprint/5635/>.
- C. Cortes and V. Vapnik. Support-vector networks. *Machine learning*, 20(3):273–297, 1995.
- T. Cover and P. Hart. Nearest neighbor pattern classification. *IEEE Transactions on Information Theory*, 13(1):21–27, 1967. doi:[10.1109/TIT.1967.1053964](https://doi.org/10.1109/TIT.1967.1053964).
- P. F. Culverhouse, R. Simpson, R. Ellis, J. Lindley, R. Williams, T. Parisini, B. Reguera, I. Bravo, R. Zoppi, G. Earnshaw, et al. Automatic classification of field-collected dinoflagellates by artificial neural network. *Marine Ecology Progress Series*, 139:281–287, 1996.
- L. S. Davis, S. A. Johns, and J. K. Aggarwal. Texture analysis using generalized co-occurrence matrices. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, PAMI-1(3):251–259, 1979. doi:[10.1109/TPAMI.1979.4766921](https://doi.org/10.1109/TPAMI.1979.4766921).
- C. de Vargas, S. Audic, N. Henry, et al. Ocean plankton. eukaryotic plankton diversity in the sunlit ocean. *Science*, 171314:29223618–25, 05 2015. doi:[10.1126/science.1261605](https://doi.org/10.1126/science.1261605).
- L. Drago, T. Panaiotis, J.-O. Irisson, M. Babin, T. Biard, F. Carlotti, L. Coppola, L. Guidi, H. Hauss, L. Karp-Boss, F. Lombard, A. M. P. McDonnell, M. Picheral, A. Rogge, A. M. Waite, L. Stemann, and R. Kiko. Global distribution of zooplankton biomass estimated by in situ imaging and machine learning. *Frontiers in Marine Science*, 9, 2022. ISSN 2296-7745. doi:[10.3389/fmars.2022.894372](https://doi.org/10.3389/fmars.2022.894372). URL <https://www.frontiersin.org/articles/10.3389/fmars.2022.894372>.
- A. Elineau, C. Desnos, L. Jalabert, M. Olivier, J.-B. Romagnan, M. Brandao, F. Lombard, N. Llopis, J. Courboulès, L. Caray-Counil, et al. Zooscanet: plankton images captured with the zooscan. 2018.
- A. Elisseeff, M. Pontil, et al. Leave-one-out error and stability of learning algorithms with applications. *NATO science series sub series iii computer and systems sciences*, 190:111–130, 2003.
- G. Gorsky, M. D. Ohman, M. Picheral, et al. Digital zooplankton image analysis using the zooscan integrated system. *Journal of Plankton Research*, 32(3):285–303, 03 2010.
- P. Grosjean, M. Picheral, C. Warembourg, and G. Gorsky. Enumeration, measurement, and identification of net zooplankton samples using the zooscan digital imaging system. *ICES Journal of Marine Science*, 61(4):518–525, 2004.
- G. Hays, A. Richardson, and C. Robinson. Climate change and marine plankton. *Trends in ecology & evolution*, 20:337–44, 07 2005.
- A. Howard, M. Sandler, B. Chen, W. Wang, L.-C. Chen, M. Tan, G. Chu, V. Vasudevan, Y. Zhu, R. Pang, H. Adam, and Q. Le. Searching for mobilenetv3. In *2019 IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 1314–1324, 2019. doi:[10.1109/ICCV.2019.00140](https://doi.org/10.1109/ICCV.2019.00140).

- Q. Hu and C. S. Davis. Automatic plankton image recognition with co-occurrence matrices and support vector machine. *Marine Ecology Progress Series*, 295:21–31, 2005.
- J.-O. Irisson, S.-D. Ayata, D. J. Lindsay, L. Karp-Boss, and L. Stemmann. Machine learning for the study of plankton and marine snow from images. *Annual Review of Marine Science*, 14(1): 277–301, 2022. doi:[10.1146/annurev-marine-041921-013023](https://doi.org/10.1146/annurev-marine-041921-013023). URL <https://doi.org/10.1146/annurev-marine-041921-013023>. PMID: 34460314.
- D. Jiang, H. Sun, J. Yi, and X. Zhao. The research on nearest neighbor search algorithm based on vantage point tree. In *2017 8th IEEE International Conference on Software Engineering and Service Science (ICSESS)*, pages 354–357, 2017. doi:[10.1109/ICSESS.2017.8342931](https://doi.org/10.1109/ICSESS.2017.8342931).
- S. R. Kerr and L. M. Dickie. *The biomass spectrum: a predator-prey theory of aquatic production*. Columbia University Press, 2001.
- R. Kiko, M. Picheral, D. Antoine, M. Babin, L. Berline, T. Biard, E. Boss, P. Brandt, F. Carlotti, S. Christiansen, et al. A global marine particle size distribution dataset obtained with the underwater vision profiler 5. *Earth System Science Data*, 14(9):4315–4337, 2022.
- A. Krizhevsky. Learning multiple layers of features from tiny images. Technical report, 2009.
- A. Krizhevsky, I. Sutskever, G. E. Hinton, F. Pereira, C. Burges, L. Bottou, and K. Weinberger. Advances in neural information processing systems, 2012.
- A. Krizhevsky, I. Sutskever, and G. E. Hinton. Imagenet classification with deep convolutional neural networks. *Commun. ACM*, 60(6):84–90, 2017.
- G. Kutyniok. The mathematics of artificial intelligence, 2022. URL <https://arxiv.org/abs/2203.08890>.
- S. Lazebnik, C. Schmid, and J. Ponce. Beyond bags of features: Spatial pyramid matching for recognizing natural scene categories. In *2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'06)*, volume 2, pages 2169–2178, 2006. doi:[10.1109/CVPR.2006.68](https://doi.org/10.1109/CVPR.2006.68).
- C. Le Quéré, R. Moriarty, R. M. Andrew, G. P. Peters, P. Ciais, P. Friedlingstein, S. D. Jones, S. Sitch, P. Tans, A. Arneeth, T. A. Boden, L. Bopp, Y. Bozec, J. G. Canadell, L. P. Chini, F. Chevallier, C. E. Cosca, I. Harris, M. Hoppema, R. A. Houghton, J. I. House, A. K. Jain, T. Johannessen, E. Kato, R. F. Keeling, V. Kitidis, K. Klein Goldewijk, C. Koven, C. S. Landa, P. Landschützer, A. Lenton, I. D. Lima, G. Marland, J. T. Mathis, N. Metzl, Y. Nojiri, A. Olsen, T. Ono, S. Peng, W. Peters, B. Pfeil, B. Poulter, M. R. Raupach, P. Regnier, C. Rödenbeck, S. Saito, J. E. Salisbury, U. Schuster, J. Schwinger, R. Séférian, J. Segschneider, T. Steinhoff, B. D. Stocker, A. J. Sutton, T. Takahashi, B. Tilbrook, G. R. van der Werf, N. Viovy, Y.-P. Wang, R. Wanninkhof, A. Wiltshire, and N. Zeng. Global carbon budget 2014. *Earth System Science Data*, 7(1):47–85, May 2015. ISSN 1866-3508. doi:[10.5194/essd-7-47-2015](https://doi.org/10.5194/essd-7-47-2015). URL <https://essd.copernicus.org/articles/7/47/2015/>. Publisher: Copernicus GmbH.
- F. Lombard, E. Boss, A. M. Waite, et al. Globally consistent quantitative observations of planktonic ecosystems. *Frontiers in Marine Science*, 6:196, 2019.
- A. R. Longhurst. Chapter 9 - the atlantic ocean. In A. R. Longhurst, editor, *Ecological Geography of the Sea (Second Edition)*, pages 131–273. Academic Press, Burlington, second edition edition, 2007.

- A. R. Longhurst and W. Glen Harrison. The biological pump: Profiles of plankton production and consumption in the upper ocean. *Progress in Oceanography*, 22(1):47–123, 1989. ISSN 0079-6611. doi:[https://doi.org/10.1016/0079-6611\(89\)90010-4](https://doi.org/10.1016/0079-6611(89)90010-4). URL <https://www.sciencedirect.com/science/article/pii/0079661189900104>.
- D. Lowe. Object recognition from local scale-invariant features. In *Proceedings of the Seventh IEEE International Conference on Computer Vision*, volume 2, pages 1150–1157 vol.2, 1999. doi:[10.1109/ICCV.1999.790410](https://doi.org/10.1109/ICCV.1999.790410).
- J. Y. Luo, J.-O. Irisson, B. Graham, C. Guigand, A. Sarafraz, C. Mader, and R. K. Cowen. Automated plankton image analysis using convolutional neural networks. *Limnology and Oceanography: methods*, 16(12):814–827, 2018.
- S. Martini, F. Larras, A. Boyé, E. Faure, N. Aberle, P. Archambault, L. Bacouillard, B. E. Beisner, L. Bittner, E. Castella, et al. Functional trait-based approaches as a common framework for aquatic ecologists. *Limnology and Oceanography*, 66(3):965–994, 2021.
- W. McCulloch and W. Pitts. A logical calculus of ideas immanent in nervous activity. *Bulletin of Mathematical Biophysics*, 5:127–147, 1943.
- R. McGill, J. W. Tukey, and W. A. Larsen. Variations of box plots. *The American Statistician*, 32(1):12–16, 1978.
- L. McInnes, J. Healy, and J. Melville. Umap: Uniform manifold approximation and projection for dimension reduction. *arXiv preprint arXiv:1802.03426*, 2018.
- R. Moriarty and T. O'Brien. Distribution of mesozooplankton biomass in the global ocean. *Earth System Science Data*, 5(1):45–55, 2013.
- E. C. Orenstein and O. Beijbom. Transfer learning and deep feature extraction for planktonic image data sets. In *2017 IEEE Winter Conference on Applications of Computer Vision (WACV)*, pages 1082–1088. IEEE, 2017.
- E. C. Orenstein, S.-D. Ayata, F. Maps, E. C. Becker, F. Benedetti, T. Biard, T. de Garidel-Thoron, J. S. Ellen, F. Ferrario, S. L. C. Giering, T. Guy-Haim, L. Hoebeke, M. H. Iversen, T. Kiørboe, J.-F. Lalonde, A. Lana, M. Laviale, F. Lombard, T. Lorimer, S. Martini, A. Meyer, K. O. Möller, B. Niehoff, M. D. Ohman, C. Pradaliere, J.-B. Romagnan, S.-M. Schröder, V. Sonnet, H. M. Sosik, L. S. Stemmann, M. Stock, T. Terbiyik-Kurt, N. Valcárcel-Pérez, L. Vilgrain, G. Wacquet, A. M. Waite, and J.-O. Irisson. Machine learning techniques to characterize functional traits of plankton from image data. *Limnology and Oceanography*, 67(8):1647–1669, 2022. doi:<https://doi.org/10.1002/lno.12101>. URL <https://aslopubs.onlinelibrary.wiley.com/doi/abs/10.1002/lno.12101>.
- T. Panaiotis, G. Boniface-Chang, G. Dulac-Arnold, B. Woodward, and J.-O. Irisson. Classification benchmark for several large plankton images datasets: Convolutional neural networks improve detection of rare taxa. in press.
- E. Parzen. On estimation of a probability density function and mode. *Ann. Math. Statist.*, 33(3):1065–1076, 09 1962.
- A. Paszke, S. Gross, F. Massa, A. Lerer, J. Bradbury, G. Chanan, T. Killeen, Z. Lin, N. Gimelshein, L. Antiga, A. Desmaison, A. Köpf, E. Yang, Z. DeVito, M. Raison, A. Tejani, S. Chilamkurthy, B. Steiner, L. Fang, J. Bai, and S. Chintala. *PyTorch: An Imperative Style, High-Performance Deep Learning Library*. Curran Associates Inc., Red Hook, NY, USA, 2019.

- F. Pedregosa, G. Varoquaux, A. Gramfort, et al. Scikit-learn: Machine learning in python. *Journal of machine learning research*, 12(Oct):2825–2830, 2011.
- M. Picheral, L. Guidi, L. Stemann, D. M. Karl, G. Iddaoud, and G. Gorsky. The underwater vision profiler 5: An advanced instrument for high spatial resolution studies of particle size spectra and zooplankton. *Limnology and Oceanography: Methods*, 8(9):462–473, 2010.
- M. Picheral, S. Colin, and J. Irisson. Webplatform “EcoTaxa”, a tool for the taxonomic classification of images., 2017. <https://ecotaxa.obs-vlfr.fr/>.
- M. Picheral, C. Catalano, D. Brousseau, H. Claustre, L. Coppola, E. Leymarie, J. Coindat, F. Dias, S. Fevre, L. Guidi, J. O. Irisson, L. Legendre, F. Lombard, L. Mortier, C. Penkerch, A. Rogge, C. Schmechtig, S. Thibault, T. Tixier, A. Waite, and L. Stemann. The underwater vision profiler 6: an imaging sensor of particle size spectra and plankton, for autonomous and cabled platforms. *Limnology and Oceanography: Methods*, 20(2):115–129, 2022. doi:<https://doi.org/10.1002/lom3.10475>. URL <https://aslopubs.onlinelibrary.wiley.com/doi/abs/10.1002/lom3.10475>.
- W. Rawat and Z. Wang. Deep convolutional neural networks for image classification: A comprehensive review. *Neural Computation*, 29:1–98, 2017.
- R. A. Redner and H. F. Walker. Mixture densities, maximum likelihood and the em algorithm. *SIAM Review*, 26(2):195–239, 1984.
- K. Saluja, A. Bansal, A. Vajpaye, S. Gupta, and A. Anand. Efficient bag of deep visual words based features to classify crc images for colorectal tumor diagnosis. In *2022 2nd International Conference on Advance Computing and Innovative Technologies in Engineering (ICACITE)*, pages 1814–1818, 2022. doi:[10.1109/ICACITE53722.2022.9823727](https://doi.org/10.1109/ICACITE53722.2022.9823727).
- R. J. Samworth. Optimal weighted nearest neighbour classifiers. *The Annals of Statistics*, 40(5): 2733 – 2763, 2012. doi:[10.1214/12-AOS1049](https://doi.org/10.1214/12-AOS1049). URL <https://doi.org/10.1214/12-AOS1049>.
- M. Sandler, A. Howard, M. Zhu, A. Zhmoginov, and L.-C. Chen. Mobilenetv2: Inverted residuals and linear bottlenecks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4510–4520, 2018.
- O. Schlimpert, D. Uhlmann, M. Schüller, and E. Höhne. Automated pattern recognition of phytoplankton—procedure and results. *Internationale Revue der gesamten Hydrobiologie und Hydrographie*, 65(3):427–437, 1980.
- C. A. Schneider, W. S. Rasband, and K. W. Eliceiri. NIH Image to ImageJ: 25 years of image analysis. *Nature Methods*, 9(7):671–675, July 2012. ISSN 1548-7105. doi:[10.1038/nmeth.2089](https://doi.org/10.1038/nmeth.2089). URL <https://www.nature.com/articles/nmeth.2089>. Number: 7 Publisher: Nature Publishing Group.
- S.-M. Schröder, R. Kiko, and R. Koch. MorphoCluster: Efficient Annotation of Plankton Images by Clustering. *Sensors*, 20(11):3060, June 2020. doi:[10.3390/s20113060](https://doi.org/10.3390/s20113060). URL <https://hal.sorbonne-universite.fr/hal-03721723>.
- D. W. Scott. On optimal and data-based histograms. *Biometrika*, 66(3):605–610, 12 1979. ISSN 0006-3444. doi:[10.1093/biomet/66.3.605](https://doi.org/10.1093/biomet/66.3.605). URL <https://doi.org/10.1093/biomet/66.3.605>.
- H. M. Sosik and R. J. Olson. Automated taxonomic classification of phytoplankton sampled with imaging-in-flow cytometry. *Limnology and Oceanography: Methods*, 5(6):204–216, 2007.

- W. G. Sprules and L. E. Barth. Surfing the biomass size spectrum: some remarks on history, theory, and application. *Canadian Journal of Fisheries and Aquatic Sciences*, 73(4):477–495, 2016.
- M. Stone. Cross-validatory choice and assessment of statistical predictions. *Journal of the Royal Statistical Society: Series B (Methodological)*, 36(2):111–133, 1974. doi:<https://doi.org/10.1111/j.2517-6161.1974.tb00994.x>. URL <https://rss.onlinelibrary.wiley.com/doi/abs/10.1111/j.2517-6161.1974.tb00994.x>.
- S. Sunagawa, S. G. Acinas, P. Bork, C. Bowler, D. Eveillard, G. Gorsky, L. Guidi, D. Iudicone, E. Karsenti, F. Lombard, H. Ogata, S. Pesant, M. B. Sullivan, P. Wincker, and C. de Vargas. Tara Oceans: towards global ocean ecosystems biology. *Nature Reviews Microbiology*, 18(8):428–445, Aug. 2020. ISSN 1740-1534. doi:[10.1038/s41579-020-0364-5](https://doi.org/10.1038/s41579-020-0364-5). URL <https://www.nature.com/articles/s41579-020-0364-5>. Number: 8 Publisher: Nature Publishing Group.
- M. Sundermeyer, Z.-C. Marton, M. Durner, M. Brucker, and R. Triebel. Implicit 3d orientation learning for 6d object detection from rgb images. In *The European Conference on Computer Vision (ECCV)*, Sept. 2018.
- X. Tang, W. K. Stewart, H. Huang, S. M. Gallager, C. S. Davis, L. Vincent, and M. Marra. Automatic plankton image recognition. *Artificial intelligence review*, 12(1):177–199, 1998.
- M. Thiel, E. Macaya, E. Acuna, et al. The humboldt current system of northern and central chile. *Oceanography and marine biology*, 45:195–345., 06 2007.
- L. Van der Maaten and G. Hinton. Visualizing data using t-sne. *Journal of Machine Learning Research*, 9:2579–2605, Nov. 2008.
- V. Vapnik. *The nature of statistical learning theory*. Springer science & business media, 1999.
- T. Volk and M. I. Hoffert. Ocean carbon pumps: Analysis of relative strengths and efficiencies in ocean-driven atmospheric CO<sub>2</sub> changes. In E. T. Sundquist, editor, *The Carbon Cycle and Atmospheric CO<sub>2</sub> : Natural Variations Archean to Present*, volume 32 of *Geophysical monograph series*, pages 99–110. ARRAY(0xc832388), Washington, D.C., 1985.
- M. Wang, X. Xu, Q. Yue, and Y. Wang. A comprehensive survey and experimental comparison of graph-based approximate nearest neighbor search. *Proc. VLDB Endow.*, 14(11):1964–1978, oct 2021. ISSN 2150-8097. doi:[10.14778/3476249.3476255](https://doi.org/10.14778/3476249.3476255). URL <https://doi.org/10.14778/3476249.3476255>.
- K. Weiss, T. M. Khoshgoftaar, and D. Wang. A survey of transfer learning. *Journal of Big data*, 3(1):1–40, 2016.
- Z. Wen, J. Shi, Q. Li, B. He, and J. Chen. ThunderSVM: A fast SVM library on GPUs and CPUs. *Journal of Machine Learning Research*, 19:797–801, 2018.
- Z. Wen, J. Shi, B. He, Q. Li, and J. Chen. ThunderGBM: Fast GBDTs and random forests on GPUs. *Journal of Machine Learning Research*, 21, 2020.
- T. Zhang. Statistical analysis of some multi-category large margin classification methods. *J. Mach. Learn. Res.*, 5:1225–1251, Dec. 2004. ISSN 1532-4435.