



**HAL**  
open science

## Results from the deep learning challenge associated with the 3MD4040 lecture on plankton classification

Jérémy Fix

► **To cite this version:**

Jérémy Fix. Results from the deep learning challenge associated with the 3MD4040 lecture on plankton classification. CentraleSupélec. 2022. hal-04549206

**HAL Id: hal-04549206**

**<https://hal.science/hal-04549206v1>**

Submitted on 17 Apr 2024

**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.

---

# Results from the deep learning challenge associated with the 3MD4040 lecture on plankton classification

---

**Jérémy Fix**  
*LORIA, CNRS,*  
*Université de Lorraine, CentraleSupélec*  
Gif-sur-Yvette, France  
jeremy.fix@centralesupelec.fr

## Abstract

This year, the deep learning lecture on the Metz campus of CentraleSupélec was evaluated by the participation to a custom challenge on plankton classification hosted on Kaggle. This paper presents the competition and the results obtained by the 13 "competing" teams. The challenge was hosted on kaggle <https://www.kaggle.com/c/3md4040-2022-challenge>.

## 1 Context

### 1.1 Why a challenge

I used to evaluate the deep learning lecture by a standard paper and pen exam. But I always thought that was not the best way to evaluate the students because deep learning is a know-how in addition to a theoretical understanding of the concepts. You cannot obtain reasonable performance with your deep learning code if you do not understand what is going on. There are so many degrees of freedom that a random exploration of the hyperparameters is hopeless.

The challenge lasted approximately 2 months, started in December 2021 and ended beginning of February 2022.

### 1.2 Grading

Grading of the students' contributions is on three topics :

- the performance on the leaderboard for 5 points
- the quality of the 20 minutes long video recorded presentation for 5 points
- the quality of the code and experimentations for 10 points

I once asked orally to provide a little article presenting the experimental track but forgot to mention it on the grading page and therefore did not ask for it at the end of the challenge.

For the performance on the leaderboard, the points were allocated as :

- 1 point if the submitted predictor performs better than a benchmark linear network
- 2 points if the submitted predictor performs better than a benchmark basic CNN network
- 2 points if the submitted predictor ranks first

### 1.3 Rules

The students were asked to respect the following rules :

- the predictor must be a neural network

- the code must be in pytorch. Using higher level framework such as pytorch lightning was allowed
- the students were not allowed to share codes between the teams
- the students were allowed and strongly encouraged to ask questions on a dedicated forum and they would be provided hints under the condition that the previous rules were not violated

In practice, video sessions were organized with the groups that had the most difficulties to come into the challenge.

## 2 The challenge on plankton classification

The data for the challenge come from the ZooScanNet dataset (<https://www.seanoe.org/data/00446/55741/>). It contains 1,433,278 images sorted in 93 taxa. We only used the images and not the native nor skimage features that were provided. It does not mean that these handcrafted features could not be useful for plankton classification, it is just the the challenge focused exclusively on image classification (but definitely, a later issue should include them).

### 2.1 Data preparation

Three folds were prepared from the original data :

- a training set of 855.394 images
- a public test set of 47.473 images
- a private test set of 47.472 images

We did not kept all the classes from the original 93 taxa. Five classes were ignored : *badfocus\_artefact*, *badfocus\_Copepoda*, *bubble*, *multiple\_\_Copepoda* and *multiple\_\_other*. Three classes were fused : *detritus*, *fiber\_\_detritus* and *seaweed*. Therefore, at the end, we get a total of 86 classes. We also kept a maximum of 200.000 per classes (before merging classes which explains why the detritus has more than 200000 samples).

One concern in preparing the data challenge is to prevent cheating such as overfitting the original dataset (which then comprise the test data). To prevent the challengers from cheating, a random rotation was applied to the image.

### 2.2 Data statistics

All the sets were identically balanced, i.e. the classes are imbalanced but imbalanced in the same proportions irrespectively of the fold. After the challenge, rereading the different classes, it might have been worth fusing artefact with detritus, and also maybe some egg classes. There are also two classes for tail and head which might gather tails and heads of different species which in no way can be considered as a single class. These two classes should have probably been discarded.

The distribution of the samples over the different folds is shown on figure 1. The dataset is imbalanced and that was one interest of the challenge. The class with the most samples is the detritus followed by the Calanoida. The class with the lowest number of samples is the Ctenophora which has only 36 samples in the training set and 1 sample in both the public and private test sets.

The images vary in size. During the preparation of the dataset, the images were resized so that their height and width do not exceed 300, keeping their aspect ratio. Although the original images vary in sizes, this choice was made to keep the weight of the whole dataset reasonable. Note however that this choice may have introduced difficulties in the recognition process (which could have been mitigated by providing the handcrafted features). Indeed, this resize erases class specific shape distributions. The distribution of heights and widths of the datasets are shown on figure 2, left for the training set, 2, middle for the test set and 2, right for the original images. For practical simplicity, only the size plot of the training set indicates the class belongings.

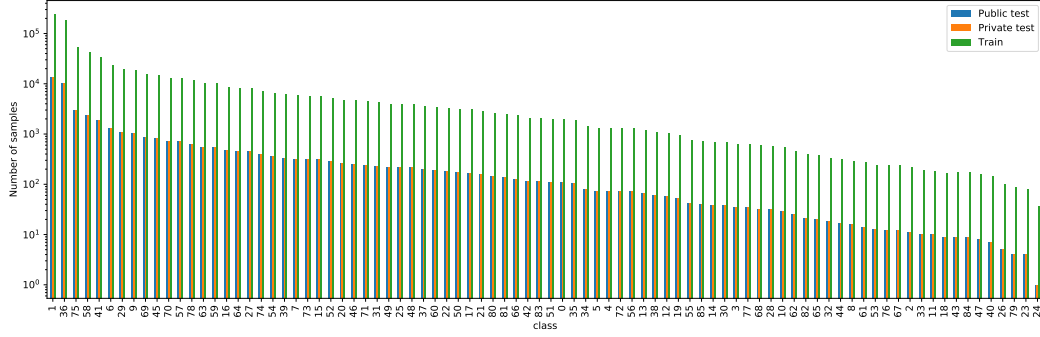


Figure 1: The distribution of samples per classes. See the appendix for the details, especially the class names

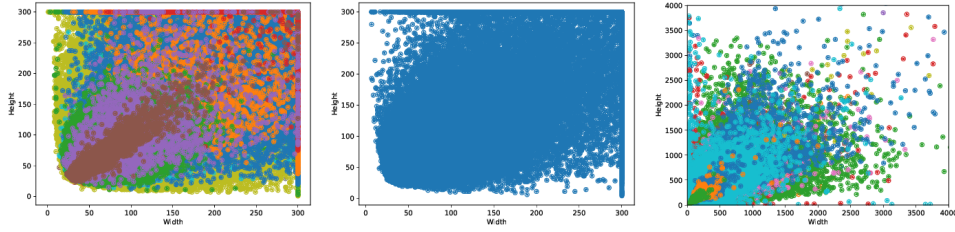


Figure 2: Distribution of the image sizes for the training set (left), test set(right) and the original dataset (right). On the plot on the right, the original sizes have been limited to 4000 for both widths and heights but there exists outliers up to 10000 in width or 12000 in height.

### 2.3 Evaluation metrics

The challenge is a classification challenge with strongly imbalanced classes. The competitors were evaluated with the macro-average F1 score which is the average of the class F1 scores.

The class F1 score is defined as the harmonic mean of the precision and recall. Given a class  $k$ , the number of true positives  $TP_k$ , false positive  $FP_k$ , and false negatives  $FN_k$ , precision, recall and F1 are computed as :

$$\begin{aligned}
 precision_k &= \frac{TP_k}{TP_k + FP_k} \\
 recall_k &= \frac{TP_k}{TP_k + FN_k} \\
 \frac{1}{F1_k} &= \frac{1}{2} \left( \frac{1}{precision_k} + \frac{1}{recall_k} \right)
 \end{aligned}$$

The macro-average F1 is then defined as the average of all the classes :

$$\text{macro-F1} = \frac{1}{86} \sum_{k=0}^{85} F1_k \tag{1}$$

## 3 The submissions of the participants

In the next section, we give a quick overview of the different techniques involved by the participants for the different topics of :

- data loading
- data augmentation

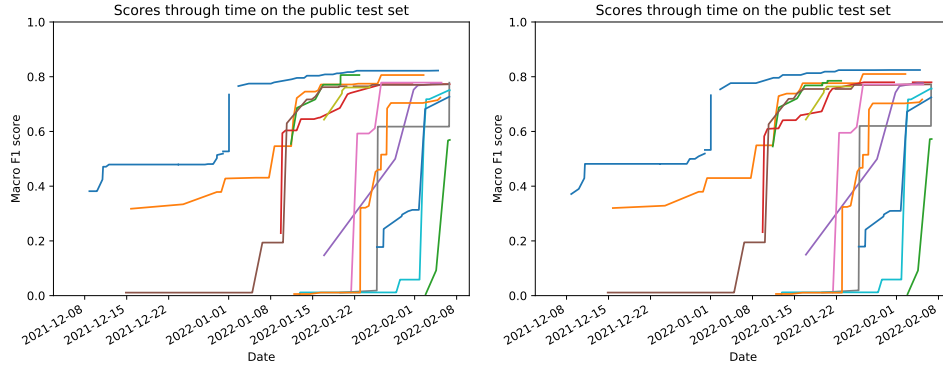


Figure 3: Macro F1 on the public (left) and private (right) test sets as a function of time.

- classification model architectures
- optimization setup (optimizer, scheduler, ..)
- handling of the class imbalance

That overview is built from inspection of the submitted codes and may not reflect all the configurations that have been tested by the participants. The collected information from the code are summarized in tables 1,2,3.

Some of the participants also considered model averaging :

SpongeBox , GrandeRegazzoni (convenet, hrnet, efficientnet)

### 3.1 The scores of the participants through time

I find it interesting to see how the submissions of the participants improved through time. The teams submitted from 5 to 79 entries and the scores they obtained on the public and private test sets are shown on figure 3.

### 3.2 Analysis of the results

The best entries of all the teams ranged from 0.71 to 0.82 (except one outlier at 0.57) on the private leaderboard in macro average F1 score. The class F1 scores, computed from the whole test data (public and private) is displayed on figure 4, left. This graph displays the distribution of class F1 scores for each team, where the teams are ordered by decreasing macro average F1.

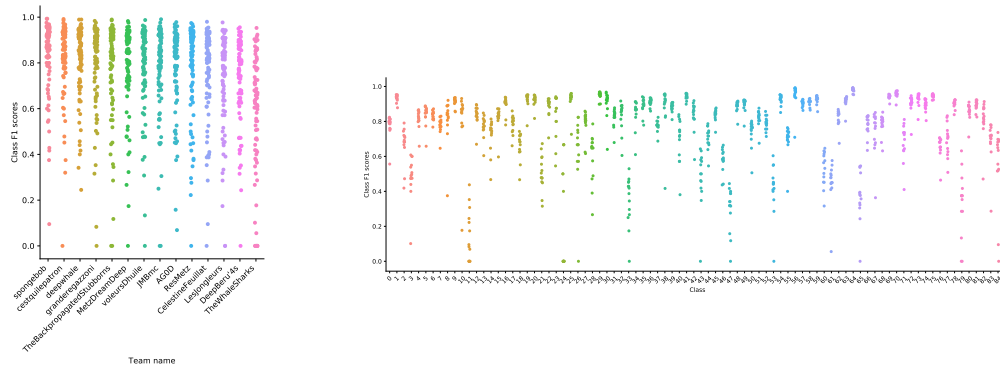


Figure 4: Class F1 scores on the test data per team (left) and per class (right).

We can also plot the F1 scores from the class perspective rather than a team perspective. That view of the class F1 scores is shown on figure 4, right. This graph plots the distribution of the teams' F1 for

Participant	Data preprocessing	Train augmentation	Test augmentation	Model architectures	Optimizer and scheduler	Class imbalance
SpongeBob	Resize(224), Normalize(0.5, 0.5)	HFlip(0.5), VFlip(0.5), Rotate(-40, 40), Translate(0.3, 0.5), Scale(0.7, 1.3), Shear(-30, 30)	None	Custom ConvNet, Pretrained torchvision ResNet, AlexNet, DenseNet121, timm ViT, EfficientNetB2, ConvNeXt, RegNetx_320, EfficientNetv2_rw_t, BeiT, CoaT	Cross Entropy loss, Adam( $lr=5e-4$ ), ReduceLROn-Plateau(validation loss), early stopping on validation F1, Batch size 64, Random train/valid split (0.95, 0.05)	Class weights in the CE loss
DeepWhale	Pad/Resize(224), ImageNet normalization	HFlip(0.5), VFlip(0.5), Rotate(-150, 150)	None	Custom CNN, pre-trained ResNet50, 101, 152, VGG19, SqueezeNet from torchvision	Cross Entropy-Loss, SGD( $1e-3$ ), ReduceLROn-Plateau(validation loss), early stopping on the validation loss, batch size 32, train/valid split (0.8, 0.2) per class	batch sampler with $f_k = 1/count_k$
GrandeRegazzoni	SquarePad(300) or ResizeCrop(128) or Resize(150)	HFlip(0.5), VFlip(0.5), Rotate(-40, 40), Translate(-0.2, 0.2)	None	Custom ConvNet, EfficientNet <sup>l</sup> , HRNet, ResNet, SwinTransformer	Cross Entropy loss, Adam( $1e-3$ ), ReduceLROn-Plateau(validation F1), early stopping on the validation loss, Batch size 128, train/valid split (0.8, 0.2) per class	batch sampler with $f_k = 1/count_k$
Voleurs d’huile	Resize(224), z-score normalization by training mean/std	HFlip(0.5), VFlip(0.5), Gaussian Blur	None	Custom CNN, pretrained torchvision ResNet and DenseNet121	Cross Entropy loss, Adam( $4e-4$ ), StepLR( $\gamma = 0.1, step = 6$ ), early stopping on the validation loss, Batch size 32, train/valid split (0.8, 0.2)	None

Table 1: Overview of the experiments of the participants. Part 1.

Participant	Data preprocessing	Train augmentation	Test augmentation	Model architectures	Optimizer and scheduler	Class imbalance
Metz Dream Deep	Resize(224), Normalize(0.5, 0.5)	Rotate(0, 360), HFlip(0.5), Translate(0.2), Normalization(0.5, 0.5)	None	Custom CNNs, resnet34, VGG16, VGG19, reg- net_y_32gf	Cross entropy loss, Adam( $1e-4$ ), Weight decay $1e-4$ , ..., batch size 128, train/valid split (0.95, 0.05)	batch sampler $f_k =$ $1/count_k$
The Backpropagated StuBBorns	SquarePad(), Resize(224), Normalize(0.485, 0.229)	torchvision AutoAug- ment	None	pretrained timm efficientnetv2_rw_s, tf_efficientnet_b8_ap, deit_tiny, cait_s24_224, coat_mini, swin transformers, tnt_s_patch16_224, dm_nfnet, resnet50 followed by 2 lin- ear/dropout/relu layers	Cross Entropy Loss, Adam( $1e-4$ ), early stopping on the vali- dation loss, batch size 32, train/valid split (0.8, 0.2)	batch sampler $f_k =$ $1/count_k$
JMBmc	Resize(224), 3 chans ImageNet Normalize	RandomRotate(30), HFlip(0.5), VFlip(0.5)	None	pretrained timm tf_efficient_b4, pre- trained torchvision, possibly frozen, fea- ture extraction with resnet50, wideRes- net50, mobilenet_v2 followed by 5 linear/dropout layers <sup>2</sup>	Cross entropy loss, SGD(0.03) with momentum, StepLR scheduler, batch size 64, train/valid split (0.8, 0.2), early stop- ping on the validation loss	None
AG0D	Resize(224), 3 chans Imagenet Normalize	RandomRotate, RandomAffine, HFlip(0.5), VFlip(0.5),	RandomRotate, RandomAffine, HFlip(0.5), VFlip(0.5),	Custom CNNs, pre- trained torchvision Resnet18, Resnet50, EfficientNetB7 ... with a random classification head <sup>3</sup>	weighted cross entropy loss, Adam(0.001), Re- duce LR on plateau, batch size 64, early stopping on the vali- dation F1, train/valid split (0.88, 0.12), per class	class weights in the CE loss $w_k =$ $\log(\sum_i count_i / count_k) >$ 0

Table 2: Overview of the experiments of the participants. Part 2.

Participant	Data preprocessing	Train augmentation	Test augmentation	Model architectures	Optimizer and scheduler	Class imbalance
ResMetz	SquarePad, Resize(224), 3 chans ImageNet Normalize	HFlip(0.3), VFlip(0.3), RandomRotate	None	torchvision pretrained Resnet18, Resnet50, EfficientNet with a random classification head	weighted cross entropy loss, Adam( $3e - 4$ ), early stopping the validation F1, train/valid split (0.8, 0.2)	class weights in the CE loss $w_k = \sqrt{(\sum_i count_i / count_k)}$ , batch sampler with the same weight as the class weights
CelestineFeuillat	Resize(224), RandAugment(num_ops=4, magnitude=7, fill=255), ImageNet normalize	None	None	LinearNet, custom CNN, torchvision pretrained resnet18, resnet50	Weighted cross entropy() loss, Adam(0.001), early stopping on validation F1, train/valid random split (0.9, 0.1), batch size 128	class weights in the CE loss $w_k = \frac{\sum_i count_i}{86count_k}$ ,
LesJongleurs	Resize(224), Normalize(0.5, 0.5)	None	None	LinearNet, torchvision pretrained resnet18, resnet50	weighted cross entropy, Adam(0.001), early stopping on validation F1, train/valid random split (0.8, 0.2)	class weights in the CE loss $w_k = \frac{\sum_i count_i}{count_k}$ ,
DeepBeru4s	Resize(256), CenterCrop(224), Resize(256), ImageNet Normalize	HFlip (0.5), Random-Rotate (-15, 15)	HFlip (0.5), Random-Rotate (-15, 15)	Linear, custom CNN, torchvision pretrained resnet34, resnet50, resnet152 with a possibly frozen backbone	weighted cross entropy loss or F1 loss <sup>4</sup> , Adam(0.001), early stopping on validation weighted cross entropy, train/valid random split (0.8, 0.2), batch size 128	class weights in the CE loss $w_k = \frac{86}{count_k}$ , batch sampler with $f_k = 1/count_k$
The Whale Sharks	Invert, CenterCrop(300) or Resize(300), Normalize(0.013,0.0546)	HFlip(0.5), Random-Rotate, Blur,	None	custom CNN, torchvision pretrained resnet101, resnet152, vgg16, vgg19	cross entropy loss or micro average F1 loss, Adam(0.001), early stopping on the batch averaged micro average F1, train/valid random split (0.8, 0.2), batch size 256	batch sampler with $f_k = 1/count_k$

Table 3: Overview of the experiments of the participants. Part 3.



Class names	Minimal F1 over all the teams
1 detritus	0.88
9 Oikopleuridae	0.85
20 Brachyura	0.82
29 Evadne	0.90
36 Calanoida	0.86
39 Noctiluca	0.81
41 Chaetognatha	0.90
54 Penilia	0.85
57 Coscinodiscus	0.86
58 Acartiidae	0.86
59 Corycaeidae	0.86
63 Limacinidae	0.80
64 cyphonaute	0.95
69 Phaeodaria	0.90
70 Ostracoda	0.89
73 nauplii Cirripedia	0.81
74 Salpida	0.80
75 Oithonidae	0.92
78 Oncaeidae	0.84
81 Cavoliniidae	0.83

Table 4: The classes for which the minimal F1 score obtained by all the teams is above 0.8.

every class. From this graph, we see that some classes are consistently well predicted (e.g. the classes 0, 1, 2, 3, 4 and 6) and some classes seem much harder to classify (80, 76, 75, 70, 66). In order to get an idea of the classes that appear to be the easiest to predict, we selected from the distribution of F1 scores, the classes for which the minimal F1 over all the teams is higher than 0.8. This choice leads to the easiest classes given in table 4.

In order to get an idea of the classes that appear to be the most difficult to predict, we selected from the distribution of F1 scores, the classes for which the maximal F1 over all the teams is smaller than 0.6. This choice leads to the hardest classes given in table 5.

The last analysis we provide is the construction of the confusion matrices for all the participants. These are given in figures 5. Note the confusion matrices have been row normalized and the figures should be understood as the fraction of a given class that has been classified as such or such class.

The confusion matrices allow to understand, when samples are misclassified, which are the most likely predicted classes. Interestingly, the top ranking entries all share similar error patterns.

First, there is a tendency to misclassify as class 1 which is the detritus class, the majority class. Second, for the top ranking entries, when there is misclassification the misclassification is not widespread over all the classes but peaky on only few classes. Indeed, although this is hard to see from the plot, the confusion matrix is very sparse.

Looking at the misclassification, it reveals that some of them would have been hard to prevent and it reveals an issue in the challenge preparation. For example, looking at the samples provided in appendix, the class 33 "Nectophore Abylopsitetrгона" seems pretty similar to the class 52 "Nectophore Diphyidae".

Class names	Maximal F1 over all the teams
11 zoea Galatheidae	0.44
33 nectophore Abylopsitetrгона	0.53
47 Euchirella	0.44
61 cirrus	0.58
65 part Copepoda	0.55

Table 5: The classes for which the maximal F1 score obtained by all the teams is below 0.6.

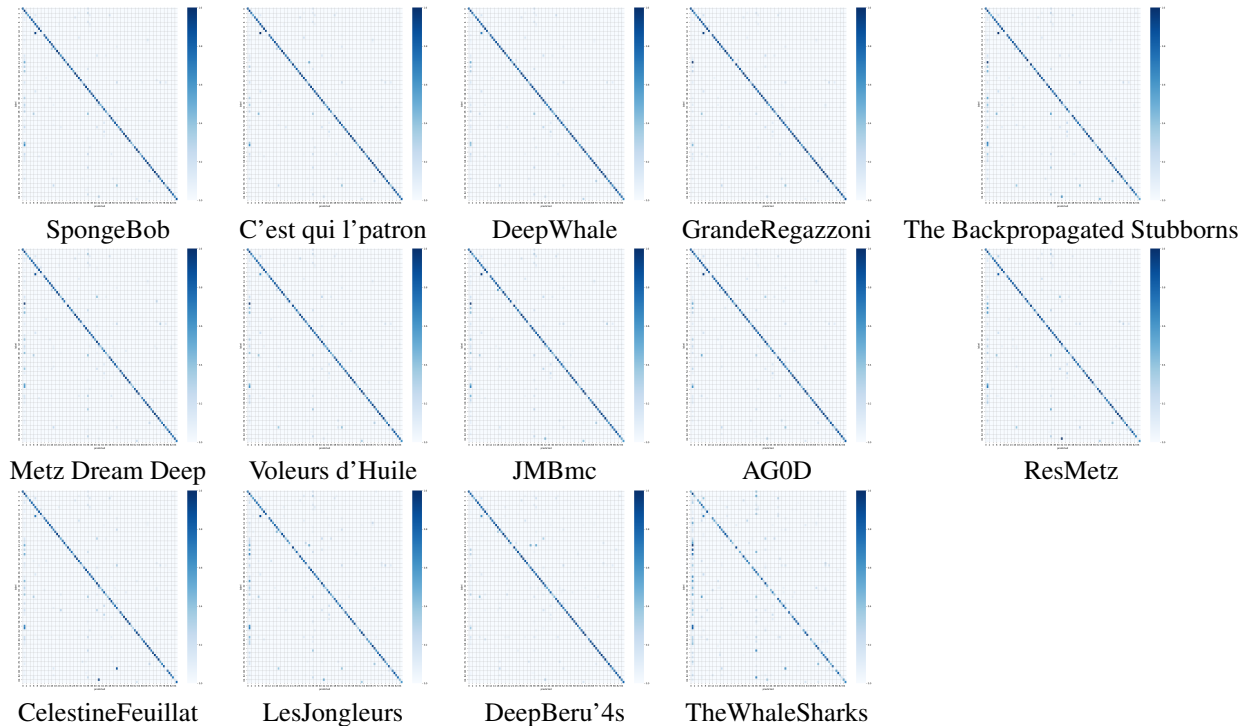


Figure 5: Confusion matrices of the best submissions of all the participants.

## 4 Sample submission

We provide a sample submission on the repository <https://github.com/jeremyfix/planktonChallenge>. This sample submission provides single models with reasonable performances. Model averaging is not performed at the time of writing.

**Preprocessing** : Every image is square padded (with white color) and resized to  $300 \times 300$ , values are divided by 255 and then normalized with mean  $\mu = 0.92$  and variance 0.16.

The training data are split in a training fold and a validation fold of 0.9/0.1. The split is stratified meaning every class has 90% of its samples in the training set and the remaining 10% in the validation set. The sample code preprocesses and splits the data once for all making later simulations faster.

**Train augmentation** : the samples can be horizontally or vertically flipped, rotated (missing pixels are filled with white). Coarse dropout, where little square regions are filled in white, is also used to mask randomly parts of the image. Examples of augmented images are shown on figure 6.

There is no test time augmentation.

**Model architectures** : several models can be trained: from basic and custom linear and convolutional architectures to architectures provided by the Timm package (resnet18, resnet50, resnet152, efficientnet\_b3, efficientnet\_b3a, regnety\_016). All the models from Timm are initialized with weights pretrained on ImageNet. The first convolution and the last fully connected layers are randomly initialized to match the dimensions of the problem (one input channel, 86 output classes).

**Optimizer and scheduler** : The training loss is a cross entropy loss. The optimizer is Adam, with an initial learning rate of 0.0003, a minibatch size of 128 or 64 (depending on the model size). The models are trained for 100 epochs (although actually training time was limited to 48 hours and large models do not perform more than 43 epochs and smaller models such as resnet18 perform 100 epochs in 35 hours).

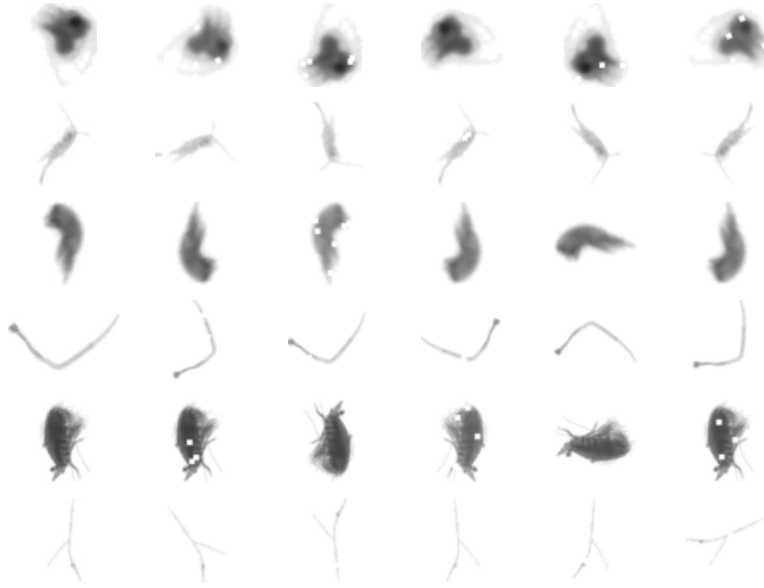


Figure 6: Examples of augmented images from the sample submission. The transforms are Horizontal flip, vertical flip, random rotation and coarse dropout. The first column represents the original images and the five following columns examples of augmented images. The little white squares on the images show the coarse dropout.

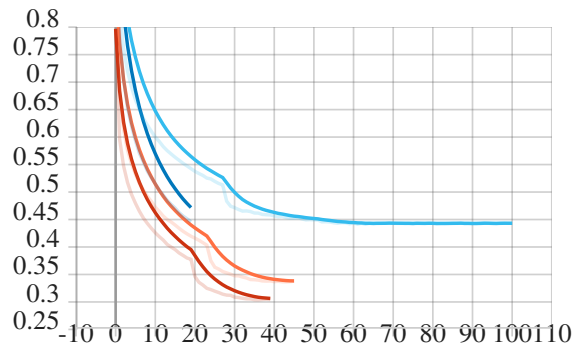


Figure 7: Averaged train cross entropy. Each were allowed to run for 48 hours. Light blue is resnet18, dark blue is resnet152, orange is regnety\_16 and red is efficientnet b3.

Mixup regularization was also introduced. In mixup, every sample  $(x, y)$  is built from two samples  $(x_1, y_1)$  and  $(x_2, y_2)$  where  $x = \lambda x_1 + (1 - \lambda)x_2$  and  $y = \lambda y_1 + (1 - \lambda)y_2$  with  $y_1, y_2$  the one-hot encoding of the classes and  $\lambda$  sampled from a  $\beta(0.2, 0.2)$  distribution.

**Class imbalance** : every minibatch is built by sampling the samples of the training folds with a weight equals to the inverse of the number of samples of the same class.

The cross entropy loss for the training and validation sets and the F1 score for the validation split are shown on figure 7-9. The confusion matrices of the best parameters for the four trained networks are shown on figure 11. For every run, the best parameters are the one with the largest validation F1 score.

The macro F1 scores on the public and private test sets of the four trained models are given in table 6. We also provide the score for the entry which predicts the class as the one maximizing the average probability of the four models. The class F1 scores for each sample submission per model and per class are displayed on the figure 10.

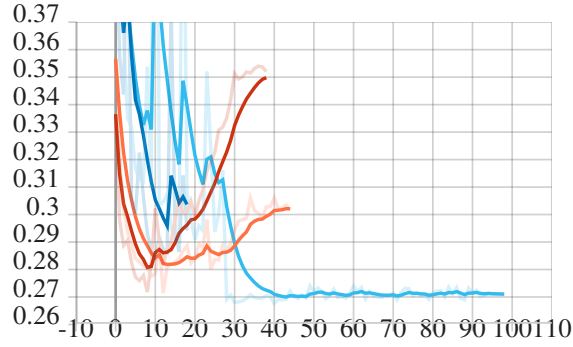


Figure 8: Averaged cross entropy on the validation fold. Each were allowed to run for 48 hours. Light blue is resnet18, dark blue is resnet152, orange is regnety\_16 and red is efficientnet b3.

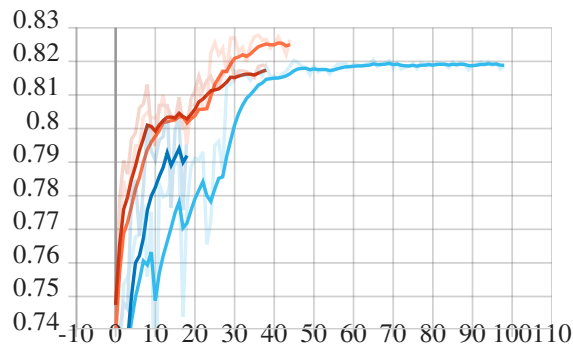


Figure 9: Macro F1 scores on the validation fold for the four trained sample submissions. Each were allowed to run for 48 hours. Light blue is resnet18, dark blue is resnet152, orange is regnety\_16 and red is efficientnet b3.

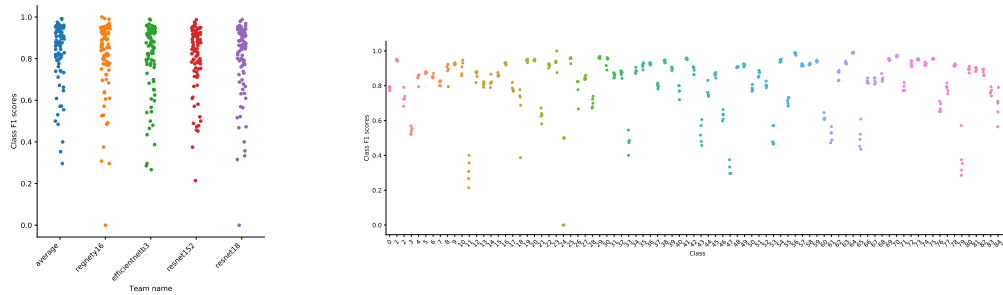


Figure 10: Class F1 scores on the test data per model (left) and per class (right).

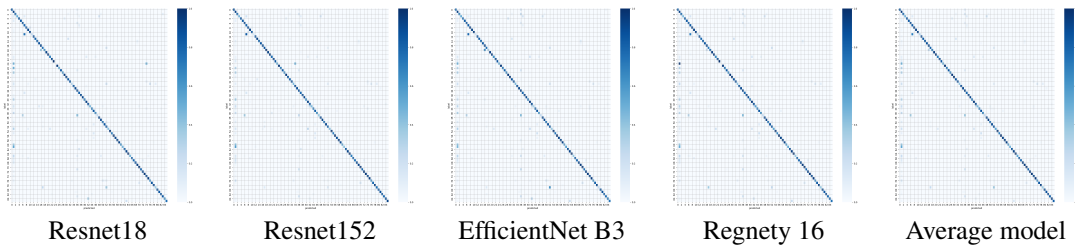


Figure 11: Confusion matrices on the validation fold of the best parameters for the four trained models.

Model	Public macro F1	Private macro F1
Resnet18	0.79723	0.79344
Resnet152	0.80178	0.79714
EfficientNet b3	0.80041	0.80489
Regnety 16	0.80178	0.79714
Average model	0.82687	0.82997

Table 6: Macro F1 scores on the public and private test sets of the challenge.

## The distribution of samples per class

The table below indicates the number of samples per class for each fold. The list is ordered from the most frequent to the least frequent classes.

Index	Name	Train	Public	Private
1	detritus	243570	13530	13530
36	Calanoida	180000	10000	10000
75	Oithonidae	53387	2965	2965
58	Acartiidae	42096	2338	2338
41	Chaetognatha	33147	1841	1841
6	Calanidae	23797	1322	1322
29	Evadne	19225	1067	1066
9	Oikopleuridae	18457	1025	1025
69	Phaeodaria	15431	857	857
45	egg other	14799	822	822
70	Ostracoda	12954	719	719
57	Coscinodiscus	12636	702	702
78	Oncaeidae	11490	638	638
63	Limacinidae	10038	557	557
59	Corycaeidae	10008	555	555
16	nauplii Crustacea	8402	466	466
64	cyphonaute	8213	456	456
27	Temoridae	8200	455	455
74	Salpida	6996	388	388
54	Penilia	6368	353	353
39	Noctiluca	6056	336	336
7	zoa Decapoda	5781	321	321
73	nauplii Cirripedia	5604	311	311
15	Foraminifera	5570	309	309
52	nectophore Diphyidae	5079	282	282
20	Brachyura	4764	264	264
46	tail Appendicularia	4609	255	255
71	Centropagidae	4391	243	243
31	Eucalanidae	4213	233	233
49	Haloptilus	3983	221	221
25	Bivalvia Mollusca	3916	217	217
48	calyptopsis	3900	216	216
37	Decapoda	3538	196	196
60	artefact	3465	192	192
22	Doliolida	3242	180	180
50	eudoxie Diphyidae	3111	172	172
17	gonophore Diphyidae	3060	169	169
21	tail Chaetognatha	2825	156	156
80	Harpacticoida	2589	143	143
81	Cavoliniidae	2517	139	139
66	Fritillariidae	2314	128	128
42	Annelida	2085	115	115
83	Euchaetidae	2086	115	115
51	egg Actinopterygii	1988	110	110

Index	Name	Train	Public	Private
0	Candaciidae	1983	110	110
35	Creseidae	1869	103	103
34	Actinopterygii	1459	81	81
5	Sapphirinidae	1328	73	73
4	Podon	1326	73	73
72	Ophiuroidea	1285	71	71
56	Pontellidae	1289	71	71
13	Rhincalanidae	1210	67	67
38	Obelia	1091	60	60
12	nectophore Physonectae	1045	57	57
19	megalopa	953	52	52
55	egg Cavolinia inflexa	757	42	42
85	Limacidae	727	40	40
14	Acantharea	686	38	38
30	Copilia	686	38	38
3	larvae Crustacea	638	35	35
77	cypris	632	35	35
68	Neoceratium	590	32	32
28	scale	576	32	32
10	Hyperidea	536	29	29
62	Luciferidae	457	25	25
82	Aglaura	388	21	21
65	part Copepoda	372	20	20
32	Pyrosomatida	336	18	18
44	Rhopalonema	308	17	17
8	Gammaridea	289	16	16
61	cirrus	269	14	14
53	head	244	13	13
76	eudoxie Abylopsis tetragona	235	12	12
67	Echinoidea	234	12	12
2	Calocalanus pavo	214	11	11
33	nectophore Abylopsis tetragona	188	10	10
11	zoea Galatheidae	185	10	10
18	metanauplii	168	9	9
43	larvae Annelida	170	9	9
84	Tomopteridae	172	9	9
47	Euchirella	156	8	8
40	Spumellaria	142	7	7
26	ephyra	98	5	5
79	gonophore Abylopsis tetragona	88	4	4
23	Scyphozoa	78	4	4
24	Ctenophora	37	1	1

**Some examples for each class**



Figure 12: Samples from the class Candaciidae



Figure 13: Samples from the class detritus



Figure 14: Samples from the class Calocalanus pavo





Figure 15: Samples from the class larvae Crustacea



Figure 16: Samples from the class Podon

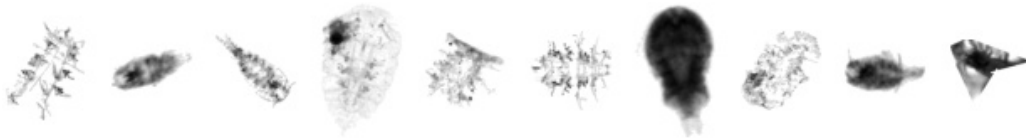


Figure 17: Samples from the class Sapphirinidae



Figure 18: Samples from the class Calanidae



Figure 19: Samples from the class zoea Decapoda



Figure 20: Samples from the class Gammaridea



Figure 21: Samples from the class Oikopleuridae



Figure 22: Samples from the class Hyperiidea

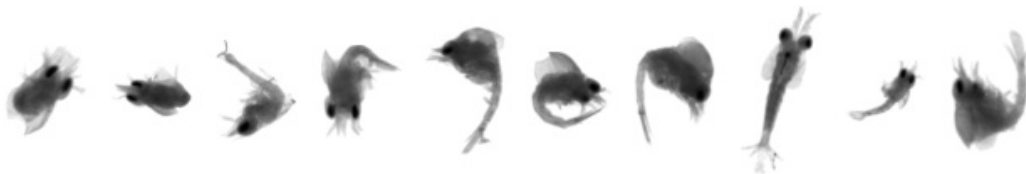


Figure 23: Samples from the class zoea Galatheidae

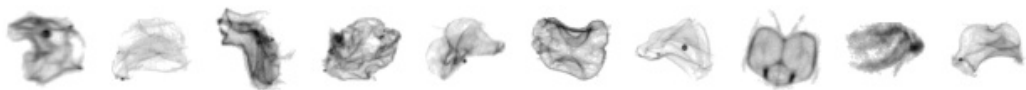


Figure 24: Samples from the class nectophore Physonectae



Figure 25: Samples from the class Rhincalanidae

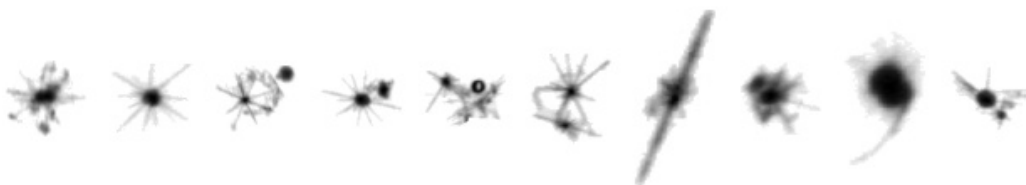


Figure 26: Samples from the class Acantharea



Figure 27: Samples from the class Foraminifera

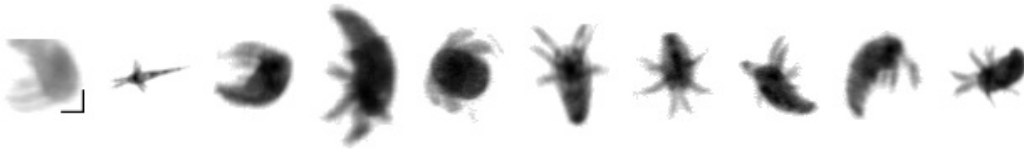


Figure 28: Samples from the class nauplii Crustacea

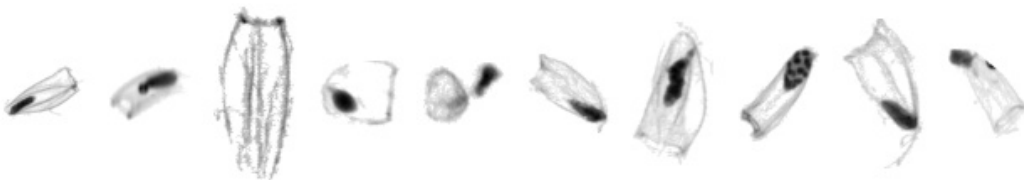


Figure 29: Samples from the class gonophore Diphyidae

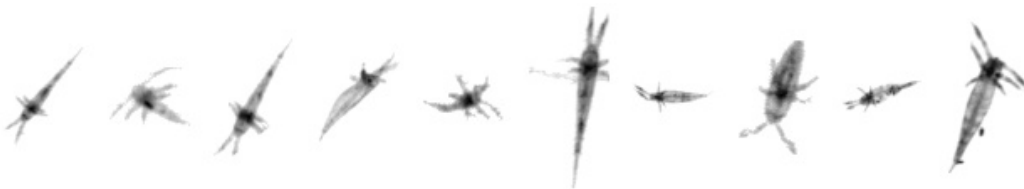


Figure 30: Samples from the class metanauplii



Figure 31: Samples from the class megalopa

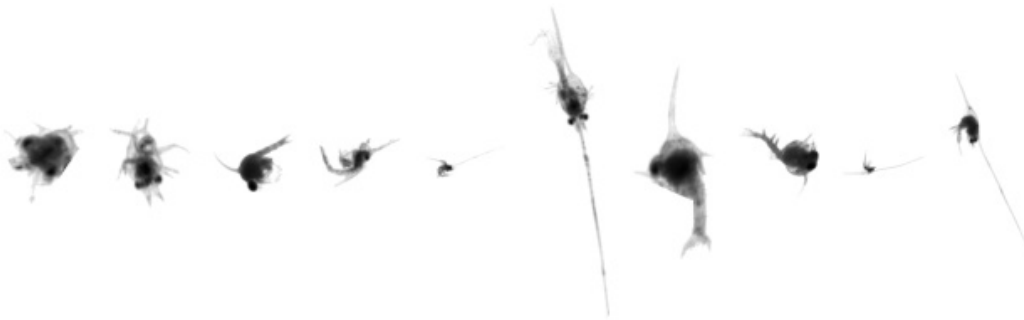


Figure 32: Samples from the class Brachyura



Figure 33: Samples from the class tail Chaetognatha

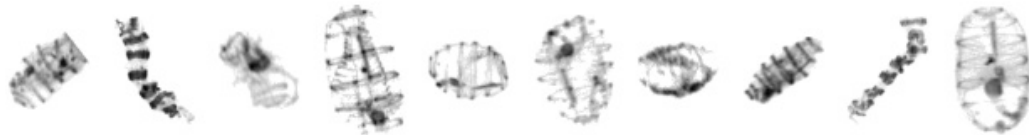


Figure 34: Samples from the class Doliolida

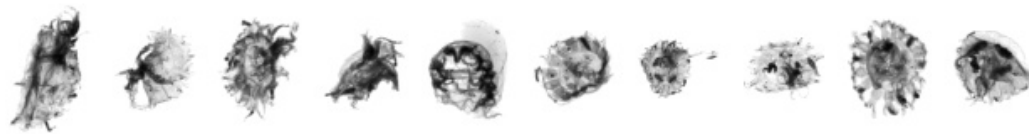


Figure 35: Samples from the class Scyphozoa

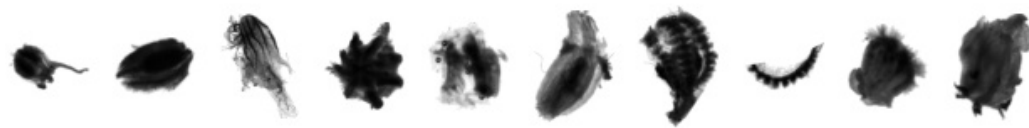


Figure 36: Samples from the class Ctenophora



Figure 37: Samples from the class Bivalvia Mollusca

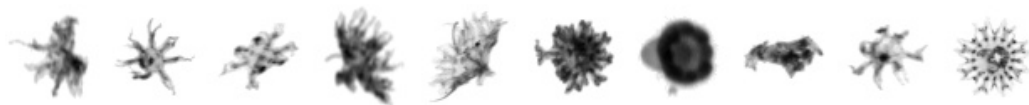


Figure 38: Samples from the class ephyra



Figure 39: Samples from the class Temoridae

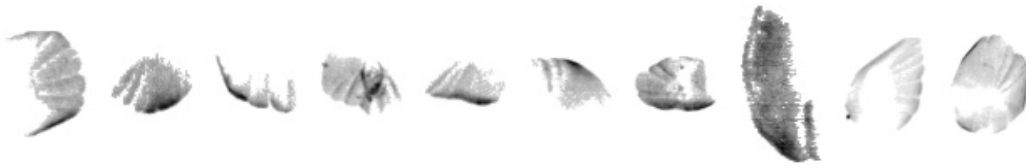


Figure 40: Samples from the class scale



Figure 41: Samples from the class Evadne

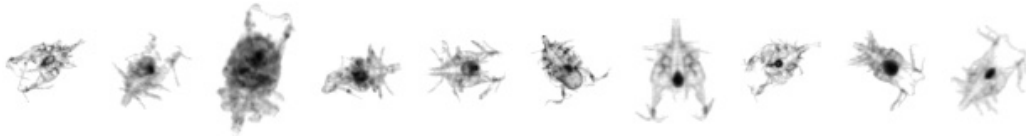


Figure 42: Samples from the class Copilia



Figure 43: Samples from the class Eucalanidae

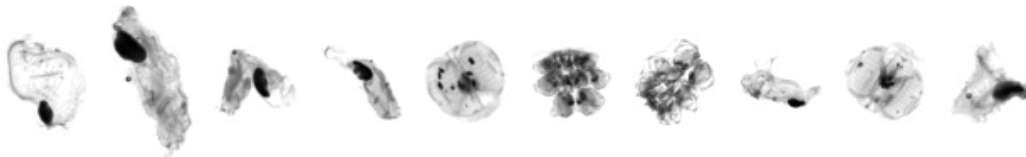


Figure 44: Samples from the class Pyrosomatida



Figure 45: Samples from the class nectophore *Abylopsis tetragona*



Figure 46: Samples from the class Actinopterygii



Figure 47: Samples from the class Creseidae



Figure 48: Samples from the class Calanoida



Figure 49: Samples from the class Decapoda



Figure 50: Samples from the class Obelia

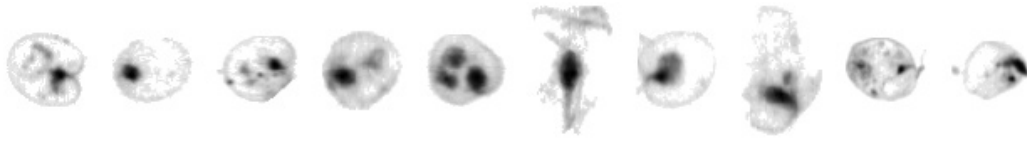


Figure 51: Samples from the class Noctiluca



Figure 52: Samples from the class Spumellaria



Figure 53: Samples from the class Chaetognatha



Figure 54: Samples from the class Annelida



Figure 55: Samples from the class larvae Annelida



Figure 56: Samples from the class Rhopalonema



Figure 57: Samples from the class egg other



Figure 58: Samples from the class tail Appendicularia



Figure 59: Samples from the class Euchirella

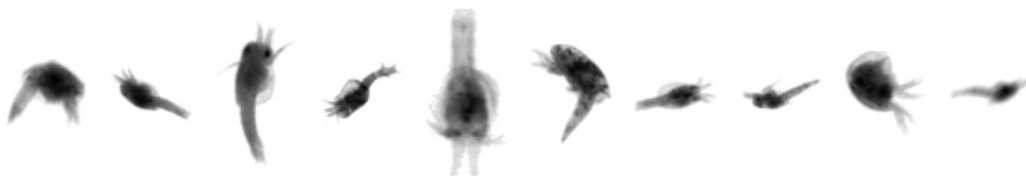


Figure 60: Samples from the class calyptopsis



Figure 61: Samples from the class Haloptilus



Figure 62: Samples from the class eudoxie Diphyidae



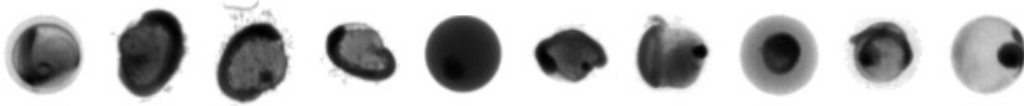


Figure 63: Samples from the class egg Actinopterygii



Figure 64: Samples from the class nectophore Diphyidae



Figure 65: Samples from the class head



Figure 66: Samples from the class Penilia



Figure 67: Samples from the class egg Cavolinia inflexa



Figure 68: Samples from the class Pontellidae

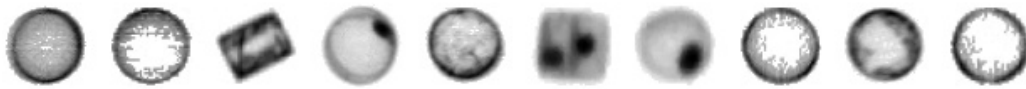


Figure 69: Samples from the class Coscinodiscus



Figure 70: Samples from the class Acartiidae



Figure 71: Samples from the class Corycaeidae

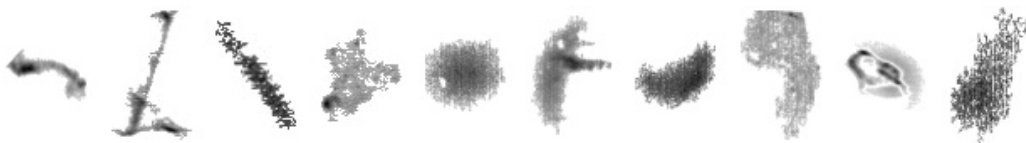


Figure 72: Samples from the class artefact



Figure 73: Samples from the class cirrus



Figure 74: Samples from the class Luciferidae



Figure 75: Samples from the class Limacinidae



Figure 76: Samples from the class cyphonaute

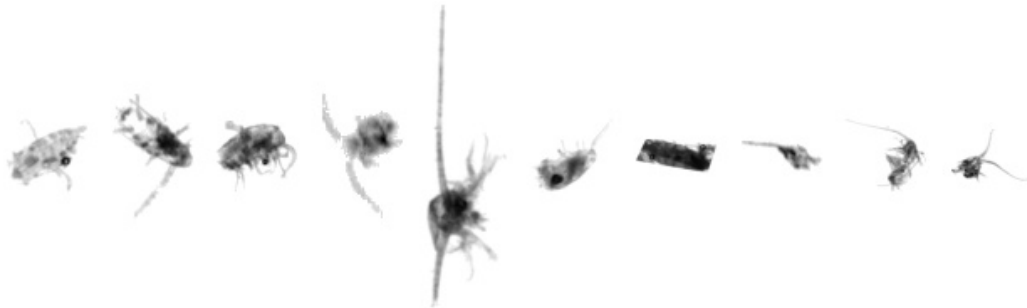


Figure 77: Samples from the class part Copepoda



Figure 78: Samples from the class Fritillariidae

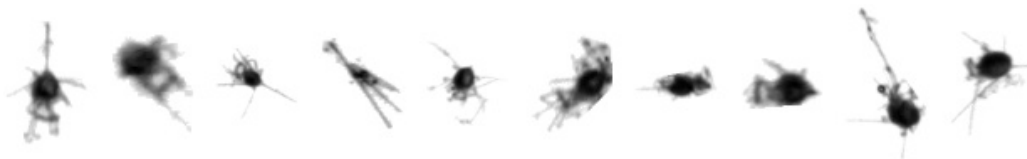


Figure 79: Samples from the class Echinoidea



Figure 80: Samples from the class Neoceratium



Figure 81: Samples from the class Phaeodaria



Figure 82: Samples from the class Ostracoda



Figure 83: Samples from the class Centropagidae



Figure 84: Samples from the class Ophiuroidea



Figure 85: Samples from the class nauplii Cirripedia

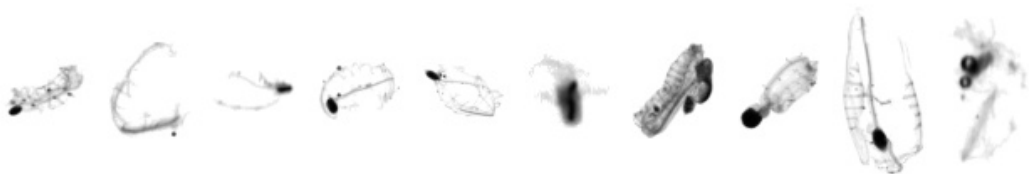


Figure 86: Samples from the class Salpida



Figure 87: Samples from the class Oithonidae



Figure 88: Samples from the class eudoxie Abylopsis tetragona



Figure 89: Samples from the class cypris



Figure 90: Samples from the class Oncaeidae

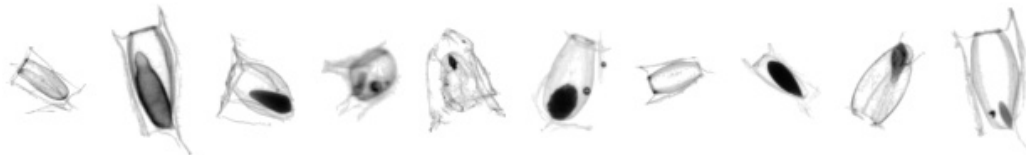


Figure 91: Samples from the class gonophore Abylopsis tetragona



Figure 92: Samples from the class Harpacticoida



Figure 93: Samples from the class Cavoliniidae

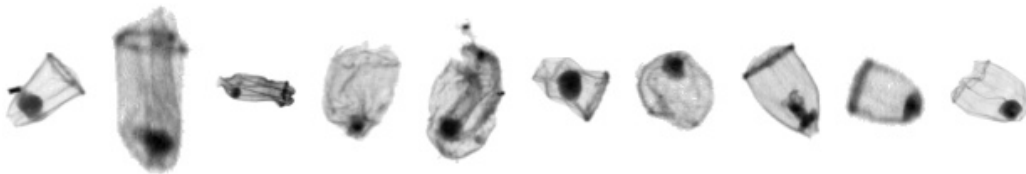


Figure 94: Samples from the class Aglaura



Figure 95: Samples from the class Euchaetidae



Figure 96: Samples from the class Tomopteridae



Figure 97: Samples from the class Limacidae