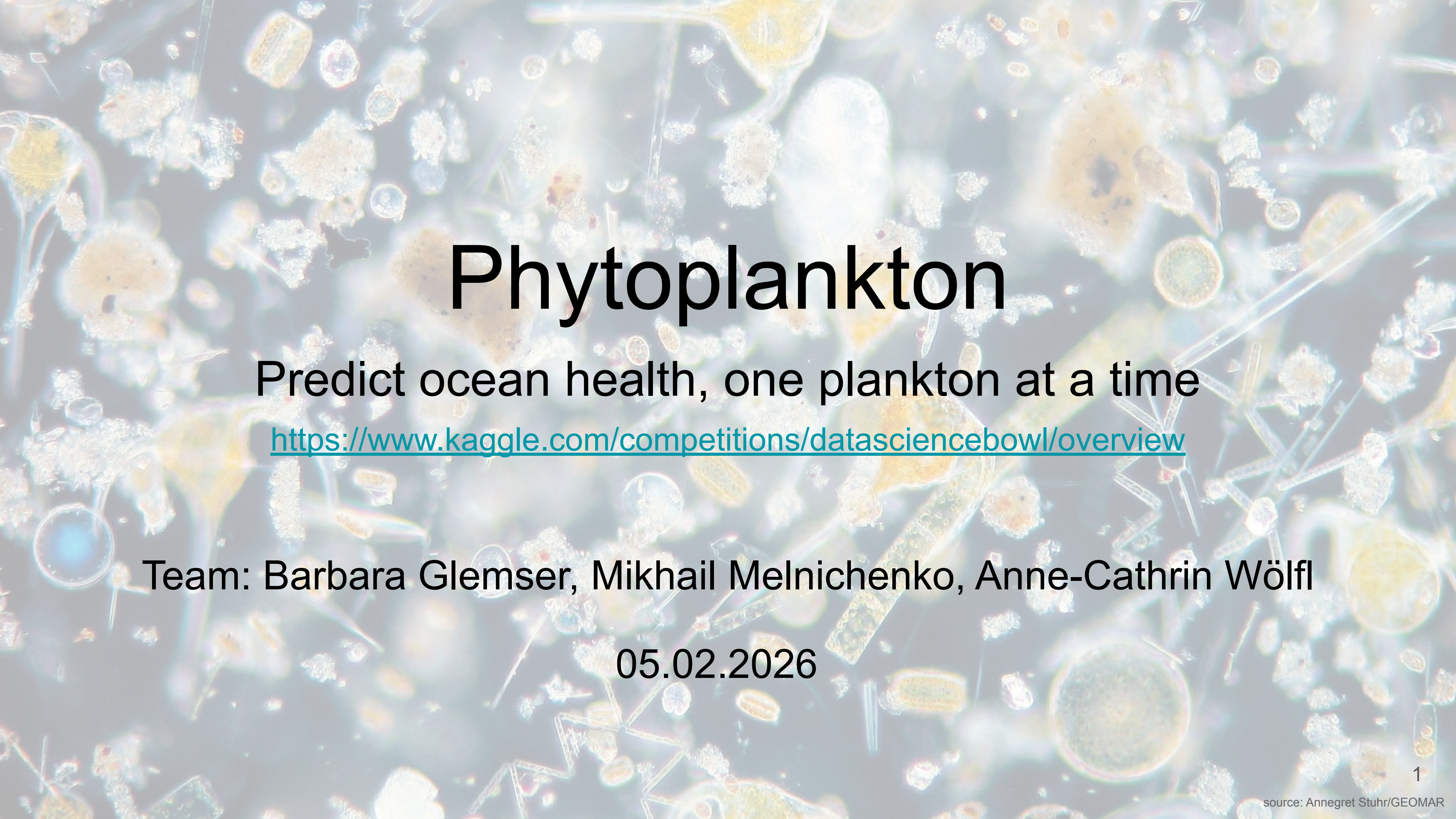


Phytoplankton

Predict ocean health, one plankton at a time

https://www.kaggle.com/competitions/datasciencebowl/overview

Team: Barbara Glemser, Mikhail Melnichenko, Anne-Cathrin Wölfl

05.02.2026

source: Annegret Stuhr/GEOMAR

Our project

- automatic classification of plankton species
- images captured May-June 2014 in the Straits of Florida

- plankton is critical for the ecosystem
- responsible for more than ½ of primary production on earth
- fixes 1/3of carbon in global carbon cycle

automated monitoring strategies/pipelines needed!

Literature review

- Classifying Phytoplankton with deep neural networks
  - blog post from winning team on kaggle
  - https://sander.ai/2015/03/17/plankton.html

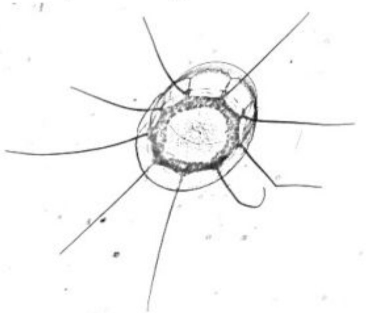
- Convolutional neural networks and vision transformers for Plankton Classification
  - https://www.sciencedirect.com/science/article/pii/S157495412500281X

- PlanktonFlow: hands-on deep learning classification of plankton images for biologists
  - https://www.biorxiv.org/content/10.1101/2025.09.19.677346v1.full

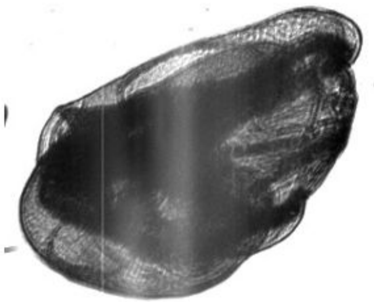
Dataset characteristics

- nearly 50 million greyscale plankton images
- 30.336 labeled images
- 121 classes
- number of images per class from 9 to 1979
- image sizes differ throughout the dataset

hydromedusae_narcomedusae

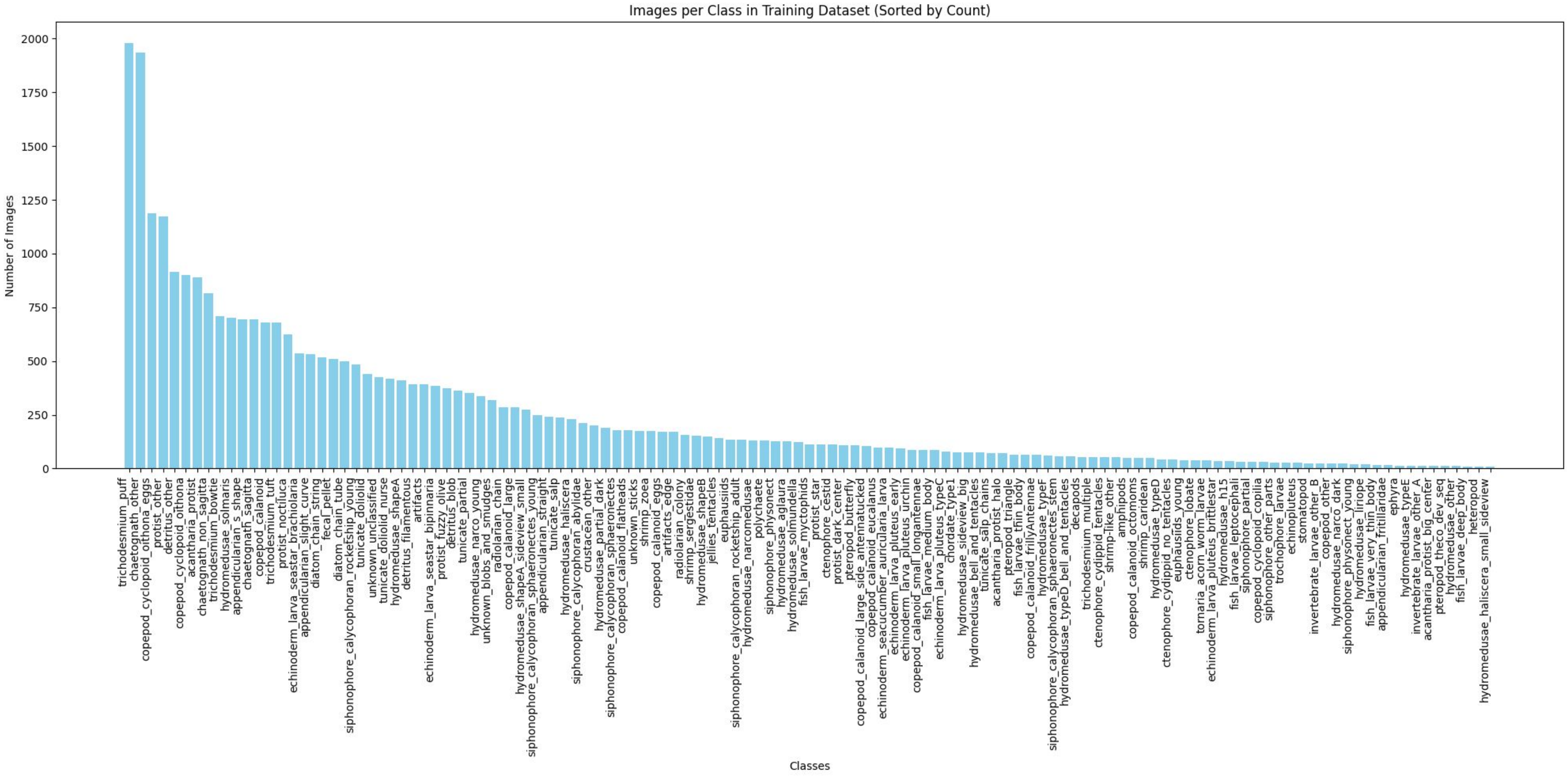


ctenophore_lobate

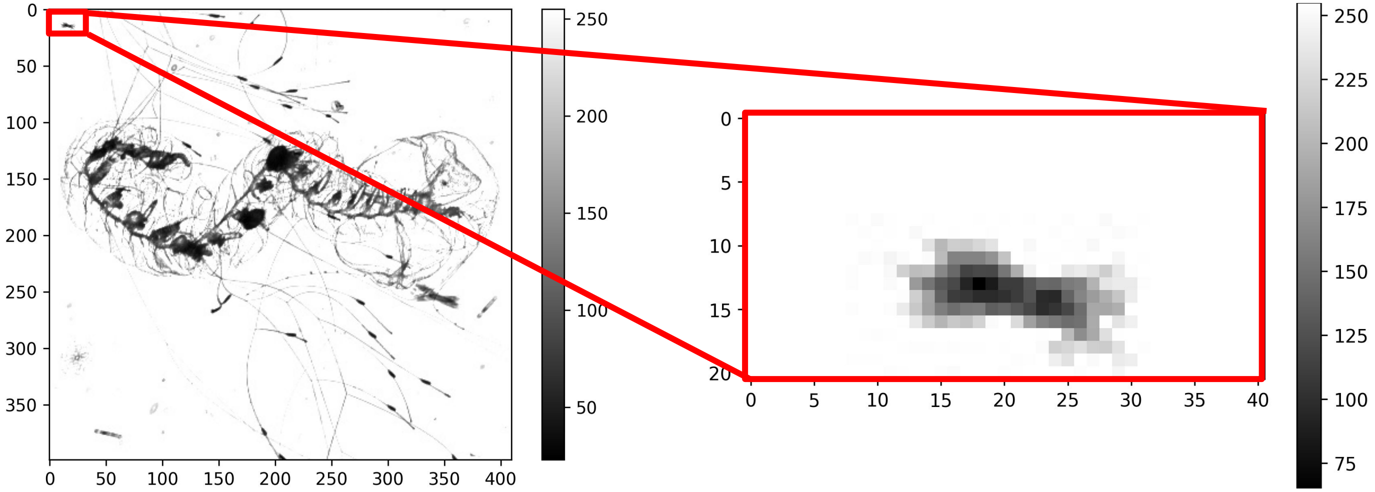


echinoderm_larva_pluteus_urchin

Dataset characteristics - imbalanced classes



Dataset characteristics - image sizes



Shape (H x W): (399, 410)
Total pixels: 163590

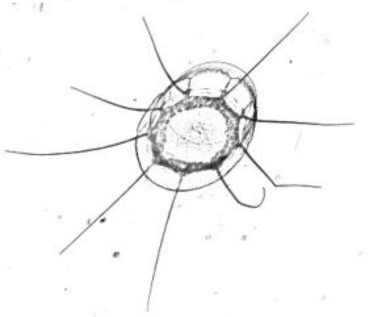
Shape (H x W): (21, 41)
Total pixels: 861

Too much upscaling > smaller images might get blurry
Too much downscaling > larger images might lose details

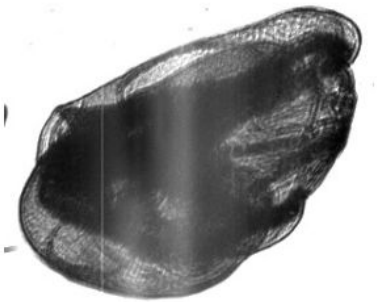
Dataset characteristics

- nearly 50 million greyscale plankton images
- 30.336 labeled images
- 121 classes
- number of images per class from 9 to 1979
- image sizes differ throughout the dataset

hydromedusae_narcomedusae



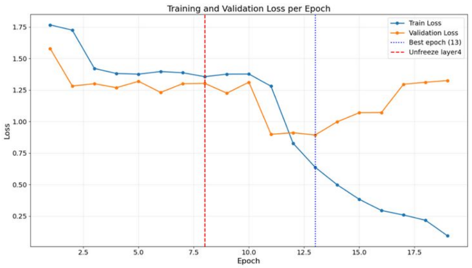
ctenophore_lobate

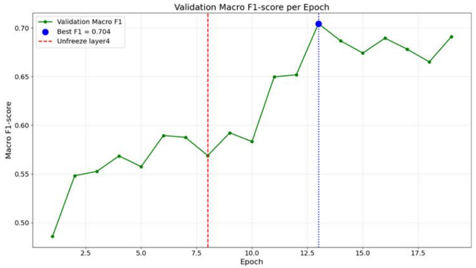


echinoderm_larva_pluteus_urchin

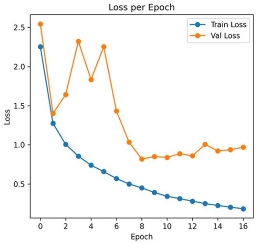
Challenges: different image sizes, unbalanced dataset, many classes including unknown classes

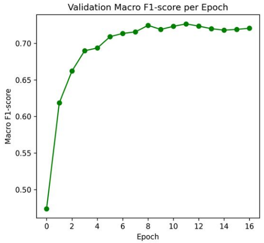
The best baseline models with only 60 classes



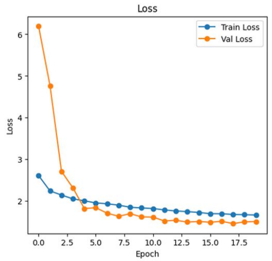


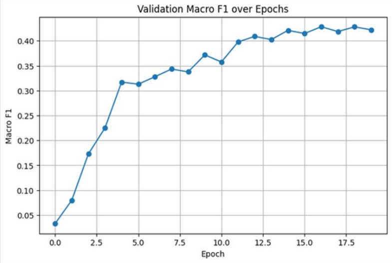
ResNet-18





EfficientNet-B0





MobileNetV2

Baseline Models

Final model

MobileNetV2 ✗

ResNet-18 ⇒ ResNet-50 ✗

DenseNet121 ✗

EfficientNet-B0 ⇒ EfficientNetV2-M ⇒ EfficientNetV2-S ⇒ **EfficientNet-B1**

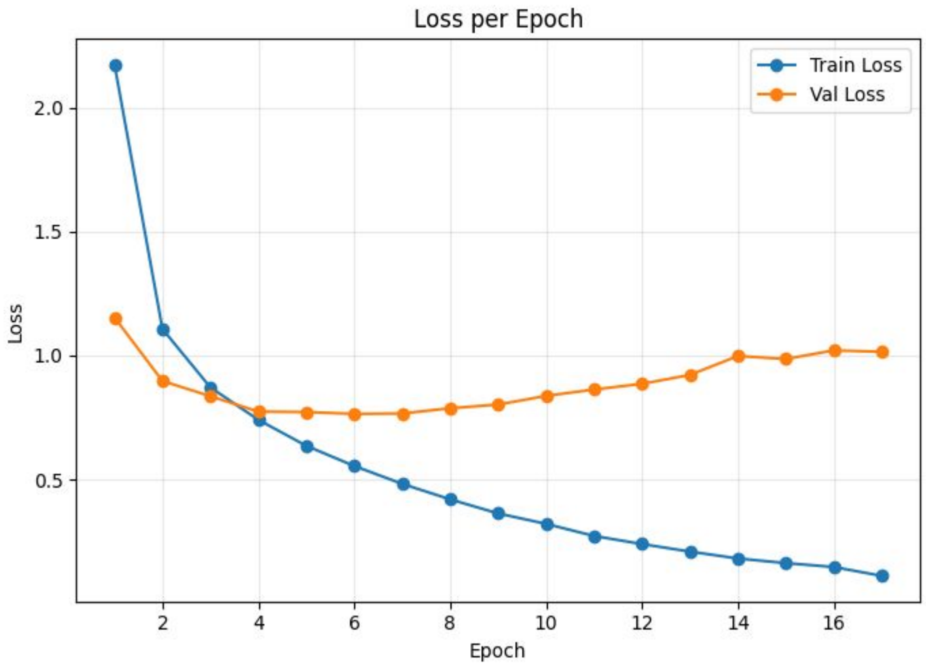
EfficientNet-B0 ⇒ EfficientNet-B1 ⇒ EfficientNet-B2 ⇒ **EfficientNet-B1**

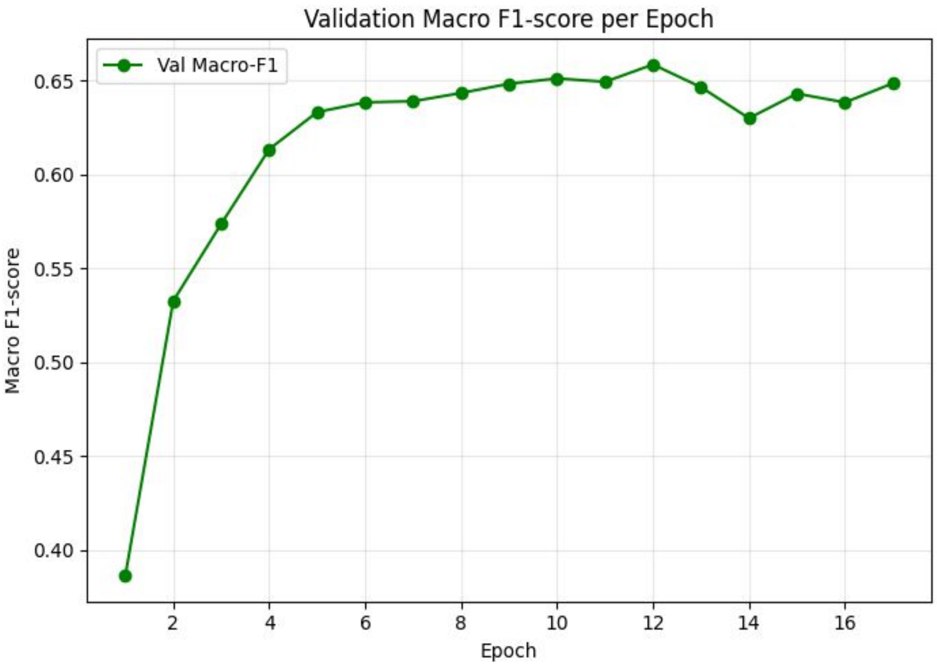
EfficientNet-B1 ⇒ EfficientNet-B1 (with unfreezing) ⇒ **EfficientNet-B1**

F1 scores from selected models

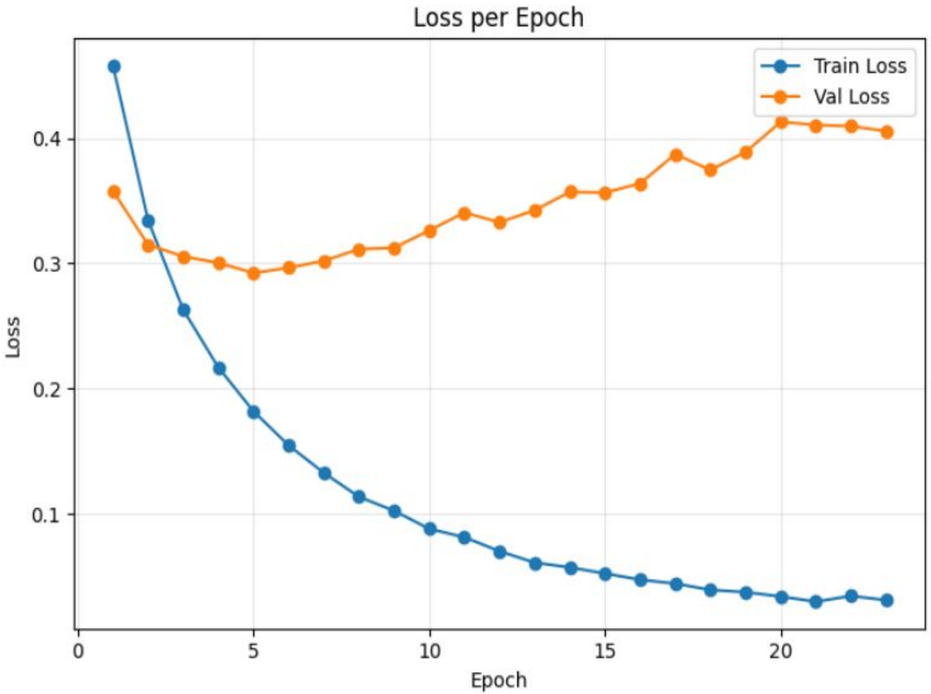
Model (selected)	Best F1 scores
MobileNetV2 (60 classes)	0.56
ResNet-18 (60 classes)	0.68
EfficientNet-B0 (60 classes)	0.72
EfficientNetV2-M (60 classes)	0.75
EfficientNetV2-S (60 classes)	0.74
EfficientNet-B0 (60 classes)	0.6
EfficientNet-B1 (60 classes)	0.765
EfficientNet-B1 version 1 (121 classes)	?
EfficientNet-B1 version 2 (121 classes)	?

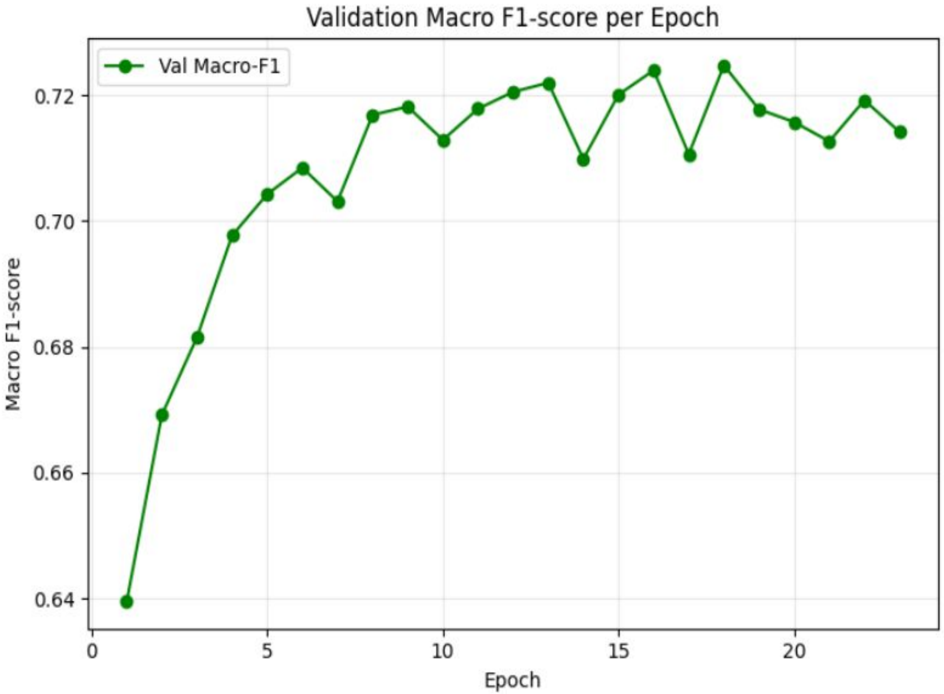
EfficientNet-B1, all classes, scheduler (version 1)



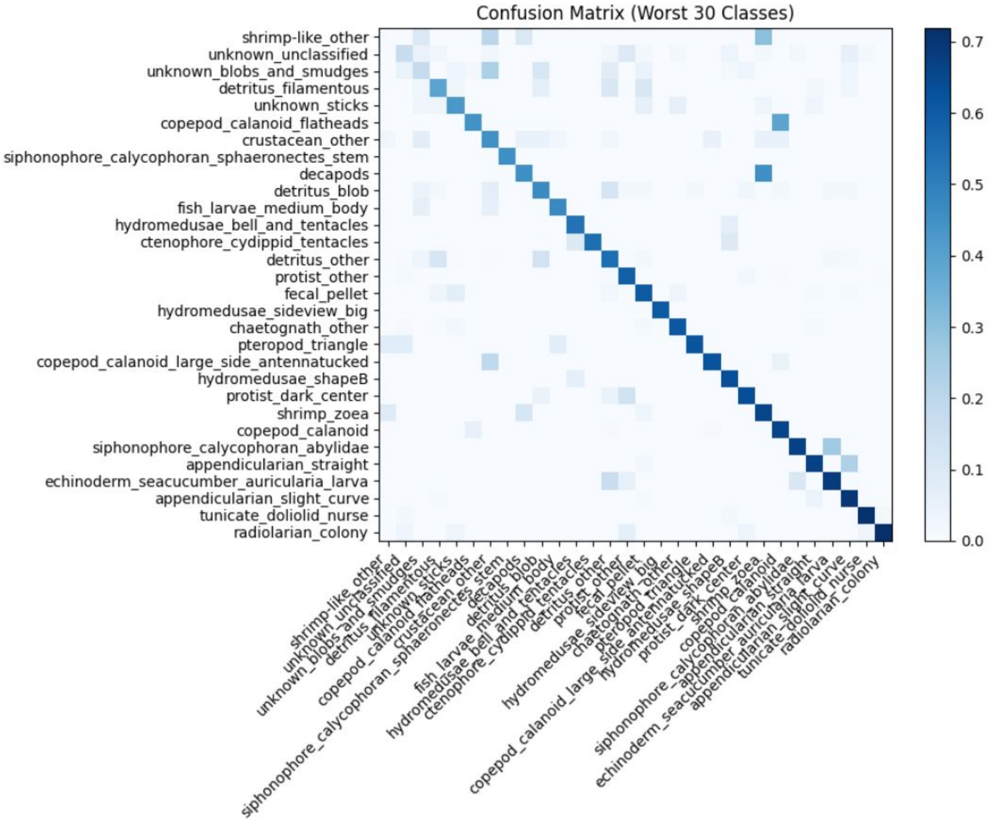


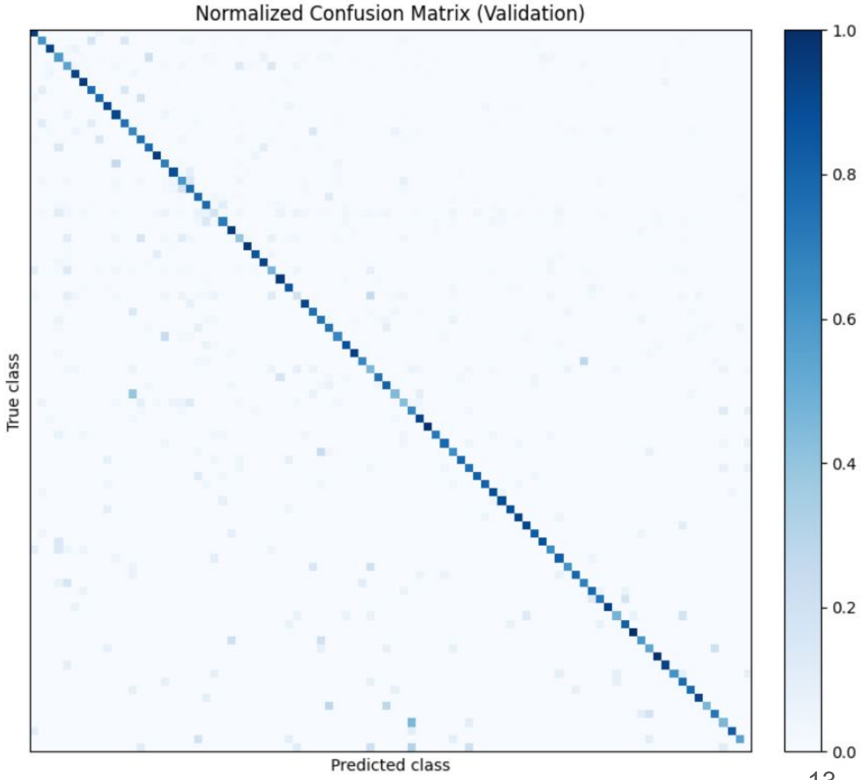
EfficientNet-B1, all classes, focal loss, scheduler (version 2)





EfficientNet-B1, all classes, focal loss, scheduler





Lessons learned

- image sizes
  - 64x64 too small, 224x224 too big -> 112x112 produced better results
- data augmentation (rotation, horizontal flip, zoom)
  - not always better results
- unfreezing
  - did produce worse results in final model
- focal loss
  - produces higher f1 but higher losses
- learning rates
  - scheduler function leads to a small improves in f1 results

Next steps

- try different augmentations for the final model
- try to unfreeze more layers and prolongate learning procedure with using of scheduler
- check a few worst classes with small number of images ‘by hand’
- uploading results in kaggle
- uploading documents and results in GitHub repo

EfficientNet-V2 small