Plankton Recognition in Images with Varying Size

Bureš Jaroslav, Eerola Tuomas, Lensu Lasse, Kälviäinen Heikki, Zemčík Pavel

in Del Bimbo, A. et al. (eds) Pattern Recognition. ICPR International Workshops and Challenges. ICPR 2021. Lecture Notes in Computer Science, vol 12666.

DOI: 10.1007/978-3-030-68780-9_11

Plankton recognition in images with varying size

Jaroslav Bureš[1,2], Tuomas Eerola[1⋆[0000−0003−1352−0999]], Lasse Lensu[1[0000−0002−7691−121X]], Heikki Kälviäinen[1[0000−0002−0790−6847]], and Pavel Zemčík[2[0000−0001−7969−5877]]

[1] Computer Vision and Pattern Recognition Laboratory, LUT University, Finland, {tuomas.eerola,lasse.lensu,heikki.kalviainen}@lut.fi
[2] Faculty of Information Technology, Brno University of Technology, Czech Republic, zemcik@fit.vutbr.cz

Abstract. Monitoring plankton is important as they are an essential part of the aquatic food web as well as producers of oxygen. Modern imaging devices produce a massive amount of plankton image data which calls for automatic solutions. These images are characterized by a very large variation in both the size and the aspect ratio. Convolutional neural network (CNN) based classification methods, on the other hand, typically require a fixed size input. Simple scaling of the images into a common size contains several drawbacks. First, the information about the size of the plankton is lost. For human experts, the size information is one of the most important cues for identifying the species. Second, downscaling the images leads to the loss of fine details such as flagella essential for species recognition. Third, upscaling the images increases the size of the network. In this work, extensive experiments on various approaches to address the varying image dimensions are carried out on a challenging phytoplankton image dataset. A novel combination of methods is proposed, showing improvement over the baseline CNN.

Keywords: Plankton recognition · Convolutional neural networks · Varying input size.

1 Introduction

Plankton are a diverse collection of organisms living in large bodies of water that are drifted by the current. They are an important part of the ecosystem as they provide the basis for the aquatic food web. Apart from this, the plankton are also the top producers of oxygen on the Earth and can be used as a good indicator of the ocean health. Therefore, monitoring plankton populations is essential. Modern imaging devices are able to produce a massive amount of plankton image data which calls for automatic solutions to analyze the data. In practice, this means recognizing the species of plankton using computer vision techniques.

⋆ corresponding author

A large amount of works on plankton recognition already exists. Recently, the majority of efforts has been put on the development of convolutional neural networks (CNN) based recognition methods that have shown to outperform traditional hand-engineering based methods with a large margin [4]. For example, in [1] a CNN architecture for plankton recognition was proposed based on the well-known VGG architecture. In [11], various CNN architectures were compared with different plankton image datasets. Moreover, different transfer learning strategies were evaluated. In [12], machine performance was compared to that of humans, and the CNN-based methods were shown to outperform the humans on the data consisting of planktic foraminifera.

The CNN based image recognition methods typically require a fixed size input. Therefore, the vast majority of existing plankton recognition methods start by rescaling the images. This, however, is not an ideal approach for typical plankton image data that are characterized with an extreme variation in both the image size and the aspect ratio (see Fig. 1). When the image is rescaled the information about the size of the plankton is lost. For human experts, the size information is one of the most important cues for identifying the species suggesting its usefulness also in automatic recognition. Downscaling images leads to the loss of fine details, such as flagella essential for species recognition. On the other hand, upscaling images increases the size of the network, resulting as longer training times and higher requirements for the amount of training data.

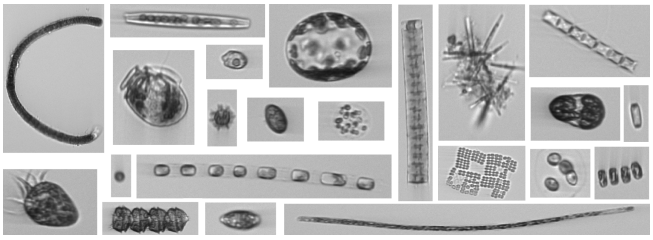

Fig. 1. Examples of plankton images with different sizes and aspect ratios.

In this paper, the problem of extreme variations in plankton image size is considered. First, existing approaches to address the varying input size on CNN-based image classification are reviewed. Then, extensive experiments on challenging plankton image data are carried out to compare the existing approaches. Finally, based on the experiments a multi-stream network utilizing a novel combination of different models is proposed.

2 CNNs with varying image size

Typical CNN architecture requires a fixed size input. In this section, existing approaches to bypass this limitation are presented.

Spatial pyramid pooling (SPP) [6] allows training of a single CNN with multiple image sizes in order to obtain higher scale-invariance and reduction of over-fitting. The convolutional and pooling layers accept feature maps of any size as they work in a sliding window manner. Limitation for input size lies in the fully connected layers, as they need an input of a fixed size. SPP accepts an input of any size and aspect ratio and produces an output of a fixed size. SPP uses a defined number of bins where each one performs pooling from one fraction of the image. For example, one bin performs pooling with the whole image (also known as global pooling), next 4 bins execute pooling with one quarter and finally 9 bins pool one ninth of the image each.

A straightforward approach to utilize the image size in the recognition is to include **the size as metadata**. This does not directly provide a solution to the need to rescale the original images but allows the recognition model to use the size information in the prediction. Various approaches to utilize the metadata in CNN-based classification models can be found in literature. Ellen et al. [5] compared several approaches for plankton recognition. Experiments on plankton images and different metadata (e.g., geometric and geotemporal data) showed that the best accuracy is achieved with the architecture with several fully connected layers after metadata concatenation. In [3], two approaches to combine image data with metadata (GPS coordinates were used in the study) were proposed. The first approach takes advantage of post-processing of an image classifier by embedding its output together with the metadata. Metadata was processed using a set of fully connected layers. After that, logits of the image classifier and metadata classifier are simply merged together. The second approach includes more interaction between the two classifiers by utilizing feature modulation.

Xing et al. [17] proposed to use **patch cropping** in a CNN-based model to recognize images with a high aspect ratio to solve the writer identification task for handwritten text. The proposed model, called Half DeepWriter takes a set of randomly selected patches cropped from the original image as an input. Furthermore, to preserve spatial information among the patches, a model, called DeepWriter was presented. This DeepWriter consisted of two Half DeepWriters. Two patches next to each other were cropped. Each patch was then supplied to one of the Half DeepWriters. These CNNs share their parameters. N pairs of patches are cropped from an input image and are fed to the model. For each pair a score vector f_i is computed and by averaging the values. The final score vector is constructed as $f_j = \frac{1}{N} \sum_{i=1}^{N} f_{ij}$.

In [13], **multi-stream CNNs** were proposed as a solution to deal with both scale-variant and scale-invariant features with CNNs. The core idea is to combine multiple CNNs and to train each one with a different input image size. The method was shown to outperform the traditional single CNN trained with images resized to a common size on the task of artwork classification. The architecture of the network was based on the ImageNet model [15] where the final average pooling layer is replaced with a global average pooling layer. Therefore, the output feature map contain the fixed size for all image scales. When applying to a new image, all softmax class posteriors from each CNN are averaged into a

single prediction. With this approach the total number of parameters is increased as the networks do not share parameters. However, the networks can be trained individually in parallel.

3 Experiments

Addressing size variation has proved to increase the accuracy of CNNs. This section provides the comparative experiments on the suitability of these approaches on plankton recognition. Four approaches are considered: SPP, metadata inclusion, patch cropping, and multi-stream networks.

3.1 Data

The data consists of phytoplankton images (see Fig. 1) and it was collected from the Baltic Sea using Imaging FlowCytobot (IFCB) [14]. The dataset contains about 33000 images labeled by a taxonomist expert into 32 different classes. The number of samples varies from 100 to 4606 per class. The images consist of one channel and their sizes are in ranges of 64 to 1276 pixels for the width and 26 to 394 pixels for the height. This variation can be considered extreme. A more detailed description of the data can be found in [2].

The data was split into 20% testing and 80% training partitions using stratified sampling. The training data was balanced so that each class contained exactly 1000 samples. If a class contained more samples, only the first 1000 images were used. If there were fewer samples then new realistic images were created through data augmentation. The following data augmentations were used: horizontal and vertical flipping, rotation of 90 degrees, scaling with the factor of 0.9 to 1.1, blurring, adjusting brightness, and adding Gaussian noise with a variance of 0.001.

3.2 CNN architectures and implementation details

To provide the baseline and to select CNN architectures for further experiments, a number of architectures were compared. For this experiment, all the images were scaled to the common size (the input size of a CNN architecture) using bicubic interpolation. To maintain the aspect ratio padding using the mean color computed from the image boundaries was used. Gaussian noise was used to reduce any artificial edges caused by homogeneous regions. Each image was normalized by subtracting the mean value from every pixel of the image and dividing the result by a standard deviation. These values were computed from the whole training set.

The following architectures were compared: *AlexNet* [10], *DenseNet121* [9], *ResNet50* [7], *MobileNet* [8], and *InceptionV3* [16], as well as, *VGG16* based models called *Al-Barazanchi* [1] and *Ellen* [5] developed especially for plankton recognition. All models were trained with the cross-validation of 10 folds with a stratified selection. The stochastic gradient descent optimizer was used, together

with the Nesterov momentum, the initial learning rate set to 0.01, the weight decay of 10^{-6} and the momentum of 0.9. For *AlexNet*, *Al-Barazanchi*, and *Ellen*, the batch size of 256 was chosen as well as training with 60 epochs. For the rest of these architectures, the batch size was set to 64 and the number of epochs to 80. The results are shown in Table 1. The deeper networks provide higher

Table 1. The baseline plankton recognition accuracy for the different architectures.

Architecture	Input size	Parameters	Accuracy
Al-Barazanchi	224×224	2 993 655	0.9341 ± 0.0025
Al-Barazanchi$_{2:1}$	316×158		0.9204 ± 0.0136
Al-Barazanchi$_{4:1}$	448×112		0.8909 ± 0.0079
AlexNet	224×224	46 854 880	0.9274 ± 0.0053
DenseNet121	224×224	7 087 607	0.9441 ± 0.0065
Ellen	128×128	885 143	0.9110 ± 0.0084
InceptionV3	299×299	21 914 903	**0.9520 ± 0.0013**
InceptionV3$_{2:1}$	420×210		**0.9525 ± 0.0033**
InceptionV3$_{4:1}$	600×150		0.9463 ± 0.0031
MobileNet	224×224	3 284 663	0.9420 ± 0.0045
ResNet50	224×224	23 694 135	0.9201 ± 0.0244

accuracy. However, this comes with the cost of longer training time. The highest accuracy of 95.20% was achieved with the *InceptionV3*. *Al-Barazanchi* obtained comparable accuracy of 93.41% with a one-third of the training time. Therefore, these two architectures were selected for further experiments.

The two architectures were further examined by training them with images of different aspect ratios to better preserve the details for images containing long plankton samples. First, the *Al-Barazanchi*$_{2:1}$ architecture was constructed for images with an aspect ratio of 2:1 where the stride of the second pooling layer was changed to (2,1). This architecture accepts images with the size of 316×158 pixels. Similarly, the *Al-Barazanchi*$_{4:1}$ architecture was constructed for images with an aspect ratio of 4:1 (448×112 pixels) where the stride of the second convolutional layer was adjusted to (4,1). Furthermore, the convolutional kernel size for the same layer was changed from (3,3) to (6,3). The same modifications were done for the *InceptionV3* architecture. To evaluate the networks, all images were flipped in such a way that the horizontal dimension was larger than the vertical dimension. The results are shown in Table 1. The architectures with the modified aspect ratio for input did not improve the results. This is as expected since the majority of the images contain the aspect ratio closer to 1:1. However, it was noticed that the modified architectures were able to correctly classify several test images for which the baseline model failed.

3.3 Spatial Pyramid Pooling

The spatial pyramid pooling layer was leveraged to enable training with images of varying resolution. This layer replaces the last pooling layer of the architecture and has a shape of $\{6 \times 6, 3 \times 3, 2 \times 2, 1 \times 1\}$ with a bin count of 50. The network was then trained with predefined image sizes (see Table 2). In one epoch both images for training and validation were resized to one of the sizes so that the whole batch consists of images with a single fixed size. After the epoch was finished, the size is switched to the next one and the process is repeated.

The *Al-Barazanchi* architecture was used for the initial experiments due to its fast training. The number of epochs was 90. The experiments with multiple different image sizes were evaluated. First, the network was evaluated with one size only (224×224) to see how the SPP layer affects the accuracy. Next, the combinations of multiple sizes (224×224, 180×180, and 256×256) were evaluated. The results are shown in Table 2. It can be seen that the SPP layer had only a minor positive effect on the recognition accuracy on its own. The same experiment was also repeated with *InceptionV3*. However, the accuracy with SPP layer (86.93%–87.61%) was considerably lower than with the baseline model (95.20 %). Therefore, the search for other size combinations was not continued.

Table 2. Accuracy for the *Al-Barazanchi* architecture using the SPP layer.

Image sizes	Accuracy
(224×224)	0.9058 ± 0.0105
(224×224), (180×180)	0.9205 ± 0.0111
(224×224), (256×256)	0.9327 ± 0.0060
(224×224), (180×180), (256×256)	$\mathbf{0.9387 \pm 0.0052}$

3.4 Metadata

The next experiment was to evaluate the effect of utilizing the size (the width and the height in pixels) of the original image as metadata. In addition, two time related features (the season and the hour) were utilized as metadata. The time metadata are motivated by the facts that there is a high seasonal variation in the plankton communities and their activity varies between the part of the day. All metadata values were normalized to $[-1; 1]$. Three architectures to include metadata proposed in [5] and [3] were examined. These are visualized in Fig. 2.

Two different approaches of training were examined. The first approach trains the whole architecture together with an embedded image model initialized with random weights. The second approach uses an image classifier that is initialized with weights loaded from a trained model and its weights are kept fixed for the time of training. Therefore, only the metadata part and the common part of

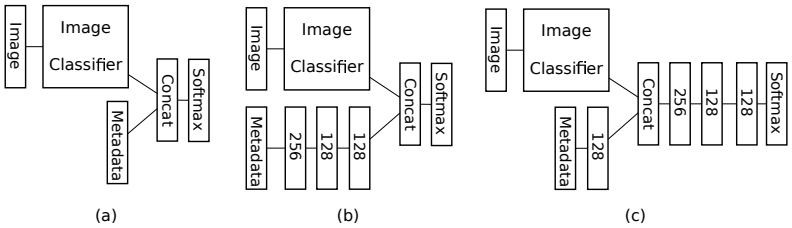

Fig. 2. Different architectures to include metadata: (a) Simple concatenation [3]; (b) Metadata interaction [5]; (c) More interaction [5].

the network are trained. The results for the *Al-Barazanchi* architecture as an image model are shown in Table 3. The best results were obtained using the pretrained image model and the Metadata interaction architecture. The effect of different types of metadata was further evaluated with the best model. While including only size information improved the accuracy over the baseline, the best accuracy was obtained using all metadata (time and size). Finally, the architecture with more interaction among metadata with both *time* and *shape* included was applied with the *InceptionV3* as the image model. The inclusion of metadata provided insignificant improvement over baseline with the accuracy of 95.22% and 95.20%, respectively.

Table 3. Accuracy for the *Al-Barazanchi* architecture with metadata.

Model	Architecture	Accuracy
	No metadata	0.9341 ± 0.0022
Blank image model	Simple concatenation	0.9392 ± 0.0037
	Metadata interaction	0.9418 ± 0.0041
	More interaction	0.9378 ± 0.0061
Pretrained image model	Simple concatenation	0.9391 ± 0.0034
	Metadata interaction (all metadata)	**0.9432 ± 0.0021**
	Metadata interaction (size)	0.9414 ± 0.0036
	Metadata interaction (time)	0.9433 ± 0.0025
	More interaction	0.9424 ± 0.0024

3.5 Patch cropping

Multiple different methods of an image patch cropping were examined. The first method uses a single patch which is randomly cropped alongside of the image. The second method uses a pair of patches to preserve spatial information between them as described in [17]. The images are padded in their width to guarantee enough space for two consecutive patches to be cropped. This pair is then supplied to the *DeepWriter* model [17]. Note that any backbone CNN

architecture can be used. The third method was to utilize the *DeepWriter* model without padding resulting in overlapping image patches for images with close to square shape. All the three methods of patch cropping are depicted in Fig. 3.

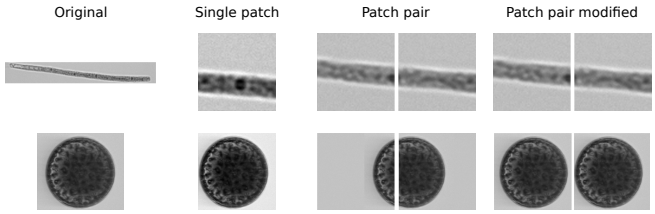

Fig. 3. Different patch cropping methods.

The first set of the experiments was carried out using *Al-Barazanchi* as the backbone architecture. Each image was first rotated into the horizontal position so that its width is greater than its height. After that, the image was resized in a way that the height of the image was the same as the height of the patch to be cropped while keeping the original aspect ratio. The model was trained for 90 epochs with a batch size of 64. The evaluation was performed through a sliding window where N patches or pairs of patches were subsequently selected from the image. Each of these patches was then evaluated by the network resulting in N prediction vectors. These vectors were finally combined by averaging them into a single prediction as described in section 2. The results are shown in Table 4.

Table 4. Accuracy for the patch cropping with the *Al-Barazanchi* architecture.

Patches	Single patch	Patch pair	Patch pair mod.
2	0.8987 ± 0.0045	**0.9298 ± 0.0030**	0.9219 ± 0.0057
4	0.9285 ± 0.0052	**0.9370 ± 0.0025**	0.9257 ± 0.0062
8	0.9301 ± 0.0050	**0.9392 ± 0.0017**	0.9276 ± 0.0063
16	0.9299 ± 0.0042	**0.9420 ± 0.0021**	0.9289 ± 0.0059

With enlarging the number of iterations, the accuracy increases. However, the time for evaluation is gradually increasing as well. While switching from 8 to 16 patches there was no significant improvement. The methods utilizing the patch pairs outperformed the single patch method which suggests that the DeepWriter architecture indeed benefits from having extra spatial information preserved by selecting two consecutive patches. The DeepWriter model outperformed the baseline model. This suggests that this method leverages small details that are being lost due to resizing. Finally, the experiment was repeated by using *InceptionV3* as the backbone architecture. This model was trained for 90 epochs with a batch size of 32. The non-modified patch pair approach was utilized for

cropping and the number of patches was set to 4. The accuracy of 95.28% was achieved which is only slightly better than the baseline.

3.6 Multi-stream CNN

To experiment with the multi-stream CNN, i.e., combining multiple CNN based models, the method proposed in [13] was utilized. Various models with different input sizes and aspect ratios were trained separately, and for the final recognition model, the prediction vectors were combined through averaging similarly to [13]. The experiment was repeated for both the *Al-Barazanchi* and *InceptionV3* architectures and the results are shown in Table 5.

Table 5. Accuracy for different model combinations. $\text{Model}_{x:1}$ stands for modification of the baseline model with the input aspect ratio of x:1.

Model combination	Al-Barazanchi	InceptionV3
Baseline ($\text{Model}_{1:1}$)	0.9341 ± 0.0022	0.9577 ± 0.0011
$\text{Model}_{1:1}$ + $\text{Model}_{2:1}$	0.9439 ± 0.0024	0.9577 ± 0.0011
$\text{Model}_{1:1}$ + $\text{Model}_{4:1}$	0.9383 ± 0.0031	0.9562 ± 0.0020
$\text{Model}_{1:1}$ + $\text{Model}_{2:1}$ + $\text{Model}_{4:1}$	0.9444 ± 0.0022	0.9596 ± 0.0005
$\text{Model}_{1:1}$ + patch cropping	0.9488 ± 0.0015	0.9580 ± 0.0023
$\text{Model}_{1:1}$ + $\text{Model}_{2:1}$ + patch cropping	**0.9499 ± 0.0018**	**0.9616 ± 0.0008**
$\text{Model}_{1:1}$ + $\text{Model}_{4:1}$ + patch cropping	0.9466 ± 0.0024	0.9606 ± 0.0002

The best improvement for *Al-Barazanchi* was found in combining it together with $\textit{Al-Barazanchi}_{2:1}$ and *DeepWriter*. This suggests that combining CNNs where each one is targeted on images with different aspect ratios can result in significant boost in accuracy. Using a method that leverages patch cropping proved to be more effective than CNNs that are fed with whole images of larger aspect ratios. The similar results were obtained also for the *InceptionV3* architecture.

3.7 Comparison of the approaches

The summary of the results for the different approaches can be seen in Table 6. For the *Al-Barazanchi* architecture every approach improved the accuracy, while the multi-stream approach provided the best accuracy. In the case of *InceptionV3* only using the multi-stream method affected significantly. This is possibly due to the already high accuracy of *InceptionV3* as well as its high complexity. It is also worth noting that while baseline accuracy is noticeably higher for *InceptionV3* compared to *Al-Barazanchi* with the multi-stream version *Al-Barazanchi* provides comparable performance. In the plankton research, image datasets are typically obtained with different imaging devices and contain different species

compositions, making it necessary to retrain the model for each dataset separately often with a limited amount of training data. Therefore, shallower models are preferred. More detailed experiments can be found in [2].

Table 6. Comparison of the accuracy for the different approaches.

Approach	Al-Barazanchi	InceptionV3
Baseline	0.9341 ± 0.0022	0.9520 ± 0.0014
SPP	0.9387 ± 0.0052	0.8761 ± 0.0153
Metadata	0.9432 ± 0.0021	0.9522 ± 0.0021
Patch cropping	0.9392 ± 0.0017	0.9528 ± 0.0009
Multi-stream	**0.9499 ± 0.0018**	**0.9616 ± 0.0008**

4 Conclusions

In this paper, various approaches to address the extreme variation in both the image size and the aspect ratio were studied for the task of plankton recognition. First, a comparison of CNN architectures was carried out. Based on the results, two architectures, *Al-Barazanchi* developed specifically for plankton recognition and considerably deeper *InceptionV3*, were selected for further experiments. Four modifications to the baseline architectures were evaluated: 1) spatial pyramid pooling, 2) metadata inclusion, 3) patch cropping, and 4) multi-stream networks. The multi-stream network combining the patch cropping model with the full image models for various aspect ratios was shown to outperform the baseline and to produce the highest accuracy for both backbone architectures. With this approach, the considerably shallower *Al-Barazanchi* architecture (3M parameters) provided comparable performance to the *InceptionV3* architecture (22M parameters), making it an attractive choice for wider use in the plankton research, characterized by a large pool of datasets with different imaging device and species compositions.

Acknowledgements

The research was carried out in the FASTVISION project (No. 321991) funded by the Academy of Finland. The authors would also like to thank Kaisa Kraft, Dr. Sanna Suikkanen, Prof. Timo Tamminen, and Prof. Jukka Seppälä from Finnish Environment Institute (SYKE) for providing the data for the experiments.

References

1. Al-Barazanchi, H.A., Verma, A., Wang, X.S.: Intelligent plankton image classification with deep learning. International Journal of Computational Vision and Robotics **8**, 561–571 (2018)
2. Bureš, J.: Classification of varying-size plankton images with convolutional neural network. Master's thesis, Brno University of Technology, Czech Republic (2020)
3. Chu, G., Potetz, B., Wang, W., Howard, A., Song, Y., Brucher, F., Leung, T., Adam, H.: Geo-aware networks for fine-grained recognition. In: ICCV Workshops (2019)
4. Eerola, T., Kraft, K., Grönberg, O., Lensu, L., Suikkanen, S., Seppälä, J., Tamminen, T., Kälviäinen, H., Haario, H.: Towards operational phytoplankton recognition with automated high-throughput imaging and compact convolutional neural networks. Ocean Science Discussions (2020)
5. Ellen, J.S., Graff, C.A., Ohman, M.D.: Improving plankton image classification using context metadata. Limnology and Oceanography: Methods **17**, 439–461 (2019)
6. He, K., Zhang, X., Ren, S., Sun, J.: Spatial pyramid pooling in deep convolutional networks for visual recognition. IEEE Transactions on Pattern Analysis and Machine Intelligence **37**(9), 1904–1916 (2015)
7. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. In: CVPR (2016)
8. Howard, A.G., Zhu, M., Chen, B., Kalenichenko, D., Wang, W., Weyand, T., Andreetto, M., Adam, H.: Mobilenets: Efficient convolutional neural networks for mobile vision applications. arXiv preprint arXiv:1704.04861 (2017)
9. Huang, G., Liu, Z., Van Der Maaten, L., Weinberger, K.Q.: Densely connected convolutional networks. In: CVPR (2017)
10. Krizhevsky, A., Sutskever, I., Hinton, G.E.: Imagenet classification with deep convolutional neural networks. In: NIPS (2012)
11. Lumini, A., Nanni, L.: Deep learning and transfer learning features for plankton classification. Ecological informatics **51**, 33–43 (2019)
12. Mitra, R., Marchitto, T., Ge, Q., Zhong, B., Kanakiya, B., Cook, M., Fehrenbacher, J., Ortiz, J., Tripati, A., Lobaton, E.: Automated species-level identification of planktic foraminifera using convolutional neural networks, with comparison to human performance. Marine Micropaleontology **147**, 16–24 (2019)
13. Noord, N., Postma, E.: Learning scale-variant and scale-invariant features for deep image classification. Pattern Recognition **61**, 583–592 (2016)
14. Olson, R.J., Sosik, H.M.: A submersible imaging-in-flow instrument to analyze nano-and microplankton: Imaging flowcytobot. Limnology and Oceanography: Methods **5**, 195–203 (2007)
15. Springenberg, J.T., Dosovitskiy, A., Brox, T., Riedmiller, M.: Striving for simplicity: The all convolutional net. arXiv preprint arXiv:1412.6806 (2014)
16. Szegedy, C., Liu, W., Jia, Y., Sermanet, P., Reed, S., Anguelov, D., Erhan, D., Vanhoucke, V., Rabinovich, A.: Going deeper with convolutions. In: CVPR (2015)
17. Xing, L., Qiao, Y.: Deepwriter: A multi-stream deep cnn for text-independent writer identification. In: ICFHR (2016)