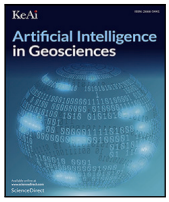
Artificial Intelligence in Geosciences 5 (2024) 100080

Contents lists available at ScienceDirect

Artificial Intelligence in Geosciences

journal homepage: www.keaipublishing.com/en/journals/artificial-intelligence-in-geosciences



The 3-billion fossil question: How to automate classification of microfossils

Iver Martinsen [a,*], David Wade [b], Benjamin Ricaud [a], Fred Godtliebsen [a]

[a] UiT - The Arctic University of Norway, Tromsø, Norway
[b] Equinor ASA, Stavanger, Norway

ARTICLE INFO

Keywords:
Self-supervised learning
Palynology
Deep learning
Microfossils

ABSTRACT

Microfossil classification is an important discipline in subsurface exploration, for both oil & gas and Carbon Capture and Storage (CCS). The abundance and distribution of species found in sedimentary rocks provide valuable information about the age and depositional environment. However, the analysis is difficult and time-consuming, as it is based on manual work by human experts. Attempts to automate this process face two key challenges: (1) the input data are very large - our dataset is projected to grow to 3 billion microfossils, and (2) there are not enough labeled data to use the standard procedure of training a deep learning classifier. We propose an efficient pipeline for processing and grouping fossils by genus, or even species, from microscope slides using self-supervised learning. First we show how to efficiently extract crops from whole slide images by adapting previously trained object detection algorithms. Second, we provide a comparison of a range of self-supervised learning methods to classify and identify microfossils from very few labels. We obtain excellent results with both convolutional neural networks and vision transformers fine-tuned by self-supervision. Our approach is fast and computationally light, providing a handy tool for geologists working with microfossils.

1. Introduction

1.1. Motivated by microfossil digitalization

Stratigraphic correlation, matching corresponding strata between different sites, wells or outcrops, is the foundation for building an understanding of the broader subsurface (Wheeler, 1958). Many techniques exist (Smith and Waterman, 1980; Baville et al., 2021), but the utilization of fossil species and assemblages of them is uniquely powerful, since apparitions, acmes and extinctions can be almost globally simultaneous on a geological time scale.

Microscopic organic remains, known as palynomorphs, are fossils that are routinely recovered and cataloged in oil & gas exploration. Analyzing palynological preparations under the microscope has historically been a manual and time-consuming process, that may be underutilized at reservoir-scale due to high expense and turnaround time. In the future, manual work will also be unsuited to low economic margin subsurface businesses such as Carbon Capture and Storage (CCS), where the understanding of reservoir connectivity and trap definition gained from determining presence/absence of stratigraphic remain crucial factors.

Recently microscope slide scanners have reached the resolution and speed needed to digitalize palynological slides. This opens an opportunity for a degree of automation in the field of palynology to those who can facilitate close collaboration between domain specialists and computer vision experts. The Norwegian Offshore Directorate (NOD) is scanning their entire slide archive, covering all exploration wells in the North Sea, Norwegian Sea and Barents Sea. These include sediments from Triassic-Neogene (and, rarely, Permian), originating in largely shallow/deep marine environments with some fluvial in the Jurassic/Triassic. The archive is estimated to number 150,000 slides - each containing on average 20,000 palynomorphs and phytoclasts. A total of 3 billion individual fossils and unavoidably extreme sparsity of labeling are challenges for applying computer-vision. In this paper, we present our methods to tackle both the huge volume of data and the lack of labels.

1.2. Deep learning and self supervised learning

Classifying or grouping images based on content using automatic algorithms is, in general, a challenging task. Even if an image classification task is conceptually simple and the subject of the photo is well defined, the placement of the subject in an image will often vary from one image to another. In addition to this, images also carry redundant information stored in the pixels surrounding the object of interest. To address these challenges, deep learning models (Goodfellow

* Corresponding author.
E-mail addresses: iver.martinsen@uit.no (I. Martinsen), dawad@equinor.com (D. Wade), benjamin.ricaud@uit.no (B. Ricaud), fred.godtliebsen@uit.no (F. Godtliebsen).

https://doi.org/10.1016/j.aiig.2024.100080
Received 7 December 2023; Received in revised form 15 April 2024; Accepted 3 June 2024
Available online 8 June 2024

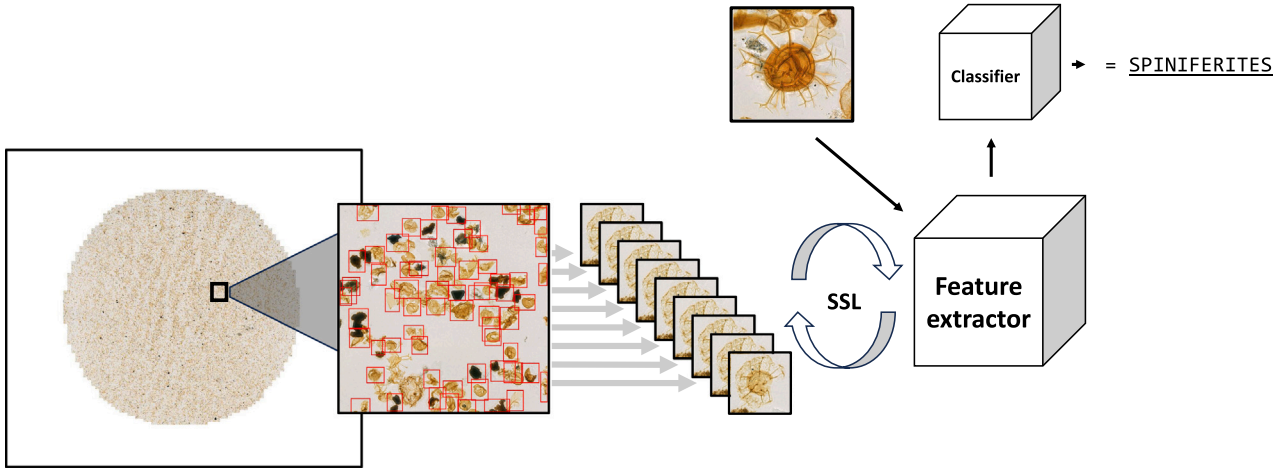


Fig. 1. From left to right: from a microscope slide, an object detection algorithm is adapted to detect individual crops (shown in with red bounding boxes). The detected crops are then used to train a feature extractor self-supervised (SSL stands for Self-Supervised Learning). Finally, the feature extractor is used on a small dataset of labeled samples to train a lightweight classifier supervised. This final classifier is trained with the features coming from the feature extractor and not directly with the images.

et al., 2016) have proven to be particularly efficient. There is a highly nonlinear relationship between images and the labels of their content, and deep learning is a framework which excels at modeling complex relationships in data. Among deep learning models, convolutional neural networks (CNNs) are designed specifically to extract useful information from images. CNNs have evolved greatly the last 10–15 years, and has until recent been the state-of-the-art in image modeling, with studies reporting excellent results (He et al., 2015; Chollet, 2017) on classifying labeled image data such as natural images from the ImageNet dataset (Deng et al., 2009). In addition to CNNs, vision transformers (ViTs (Dosovitskiy et al., 2021)) - which are inspired by large language models, have emerged as a competitive model design for image classification. Today, both CNNs and ViTs are commonly used for image classification, and there is no clear winner or best architecture for this task (Guo et al., 2022; Raghu et al., 2021). Hence, it is standard to test and compare both architectures, and that is what we do here.

Most of the deep learning literature has up until recently been focused on classification tasks with labeled data (supervised learning). However, real-world applications often involve data in which the number of labels and expert annotations is small compared to the sample size of the data. Motivated by this, self-supervised learning (SSL) is a learning paradigm that has gained popularity in recent years. In contrast to supervised learning where the aim is to map an input to its label, SSL incorporates a different learning objective that does not require any labels. Instead, SSL methods construct a task that a network should solve (a "pretext" task) based solely on the samples. The main idea is to take each sample in the dataset, modify it in some way that does not remove the important information and teach the deep network to identify both the original and the modified sample as (close to) identical in its inner representation. Examples of pretext tasks used for SSL include adding noise (Vincent et al., 2008) or masking areas in images (He et al., 2021). The inner representation, also called latent representation or embeddings, is a vector of values that contain valuable information for discriminating the samples. In a second step, this inner representation is used for unsupervised classification of the data or classification with a simple method such as logistic regression or k-nearest neighbors. The latent representation concentrates the important information contained in samples and makes the classification task easier, requiring much fewer labels for training. The SSL deep network becomes a "feature extractor" for the downstream task of classification.

In our work, we have focused on methods that incorporate augmentation-based strategies, which is a family of SSL methods that have been shown to be particularly promising for images. We make use of the popular SimCLR (Chen et al., 2020a) and DINO (Caron et al., 2021) strategies.

1.3. Our contribution

We propose an automatic pipeline for microfossil extraction and classification from raw microscope pictures. The method is fast and efficient and does not require intensive computing power. We show that our approach improves the state-of-the-art for fossil extraction. The identification of individual species with machine learning is new and promising.

Our approach, outlined in Fig. 1, has two main steps:

1. **Microfossils detection.** The first challenge is detecting individual microfossils from the microscope images. We provide a comparison of pipelines for automatically detecting individual microfossil crops from high-resolution whole-slide images completely without expert annotations.
2. **Microfossil identification.** The second challenge is to correctly associate each fossil detected with a genus/species in an accepted taxonomy. For this task, we evaluate the performance of SSL methods to train convolutional neural networks (CNNs) and vision transformers (ViTs), which pass image embeddings to simple downstream classifiers that may be trained on suitable taxonomic subsets.

2. Materials and methods

2.1. Data

The digital slides used for analysis were provided by the Norwegian Offshore Directorate (NOD), accessible through Diskos,[1] the Norwegian National Data Repository for Petroleum data. These consist of 215 whole slide images (WSIs) of palynology slides from a single wellbore (NO 6407/6-5[2]) in the Norwegian Sea.

The microfossils used in our analysis are produced mainly from drill cuttings samples and a few from core samples, taken at different depths covering mainly marine, deltaic/shoreface environments from Early Jurassic to Early Miocene. The rock samples are crushed into fine-grained particles, before being exposed to a chemical procedure that effectively removes any non-organic material (Halbritter et al., 2018). The organic remains are microfossils that are put on a microscope slide.

[1] see https://www.sodir.no/en/diskos/.

[2] https://factpages.sodir.no/pbl/wellbore_documents/3921_6407_6_5_COMPLETION_REPORT.pdf.

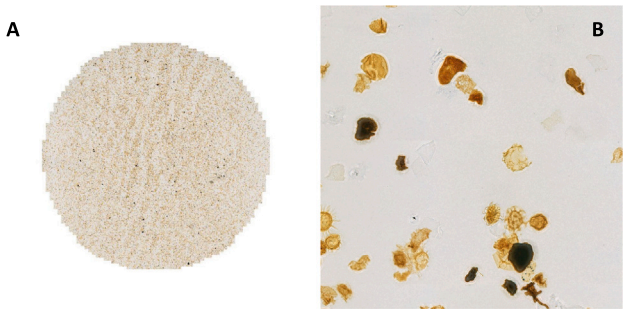


Fig. 2. Microfossil data. A microscope whole slide image of organic microfossils. The whole slide image is stitched together by multiple tiles. A: Thumbnail of a single microscopic slide. The non-empty region of the image consists of approximately 40 x 40 = 1600 tiles. B: Single tile (2048 × 2048 pixels) from a microscopic slide.

Typical size of palynomorphs are in the range 5–500 μm.[3] Finally, the microscope slides are digitized using a 3DHISTECH PANNORAMIC 1000 high-resolution scanner, with a pixel size of 0.25μm. The 215 slides correspond to the measured downhole depths of 1200 m to 2760 m. Fig. 2A shows a thumbnail of a single slide which itself consists of multiple smaller tiles (Fig. 2B). Based on applying our detection method to many slides from several wells the number of fossils in each slide is observed to be in the range of 10,000 to 50,000, implying there will be ∼2–10 million from the slides we have available in this well. Except for a few labeled images described below, the data is unlabeled and contains no information about the coordinates or types of objects (species) present in the slides.

In addition to the unlabeled whole-slide images, the data also include a sample of labeled crops drawn largely from shallow marine settings in the Cretaceous/Paleogene. This relatively tiny subset of fossils (1123 images) was selected and labeled by an expert palynologist, and the labeled set totaled 246 distinct species. There was, however, some uncertainty related to the correctness of the species for a number of the images. We assume that the actual number of species present in the slides far outnumbers the number of labeled species. The role of the labeled crops is to be used to evaluate the models trained on unlabeled slides. Due to the small number of labels for some of the species, we decided to further merge groups that belonged to the same genus, resulting in a data set containing 123 different genera (see Fig. 3 for an example). Table 1 shows the 20 most abundant genera in the labeled set. 45 of the genera contained only a single labeled example (see Appendix A, Fig. 10 for a bar chart of the total count for each genus).

2.2. Extracting microfossils from the microscope image

Our approach to microfossil analysis involves the detection of fossils from the slide and the subsequent classification of the individual crops. Object detection models such as YOLO (Redmon et al., 2016) and R-CNN (He et al., 2018) are designed to do this in an end-to-end fashion, however, they both require annotations for both the object coordinates and the object label and are thus not suitable for our problem. As our approach in model training is completely self-supervised, we instead adopt a two-stage approach where the first step is to construct bounding boxes for the individual crops in the slide. These crops will be used for self-supervised training in the second stage. As a first step, we therefore need to generate rectangular bounding boxes enclosing each fossil in a slide before extracting the crops thereafter. Due to the vast amount of fossils in the data, the bounding boxes needs to be obtained in an automatic fashion. The automatic method needs to be fast, while simultaneously ensuring that the crops are enclosed appropriately. For this

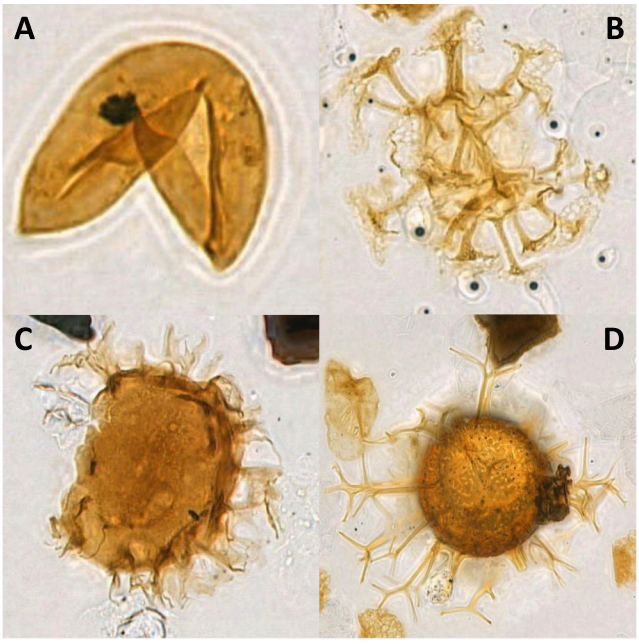


Fig. 3. Labeled crops. Examples of labeled crops. The description refers to the original annotation. The genus which was used for the class name is shown in parenthesis. A: *Inaperturopollenites hiatus* (*Inaperturapollenites*). B: *Areosphaeridium diktyoplokum* (*Areosphaeridium*). C: *Glaphyrocysta sp.* (*Glaphyrocysta*). D: *Spiniferites manumii* (*Spiniferites*).

Table 1
Classes The labeled data consists of 1123 fossils from 123 different genera. This table lists the 20 genera with the most labels. The majority of the labeled crops belong to one of these genera.

Class	Class name	#species	Class count
0	alisocysta	1	25
1	areoligera	6–7	65
2	areosphaeridium	2	28
3	azolla	1	22
4	bisaccate	1	25
5	cleistosphaeridium	2	26
6	deflandrea	6	43
7	eatonicysta	1	33
8	glaphyrocysta	2–4	26
9	hystrichokolpoma	2–3	23
10	hystrichosphaeridium	2	22
11	inaperturopollenites	1	50
12	isabelidinium	2	23
13	palaeocystodinium	5	69
14	palaeoperidinium	1	39
15	phthanoperidinium	5	62
16	spiniferites	10 or fewer	32
17	subtilisphaera	2–3	25
18	svalbardella	3	30
19	wetzeliella	5	29

application we demonstrate two different approaches. The first method is based on image analysis techniques used in recent works in geology, while the second method is based on a machine learning pretrained object detection method. The methods are compared qualitatively; the machine learning method shows superior quality in extracting single fossils.[4]

Standard image processing approach. The image analysis preprocessing method is based on recent work in geology (Johansen et al., 2021; Johansen and Sørensen, 2020). The pipeline consists of Gaussian

[3] see https://palynology.org/what-is-palynology/palynomorphs/.

[4] Code available at github.com/IverMartinsen/scampi-preprocessing.

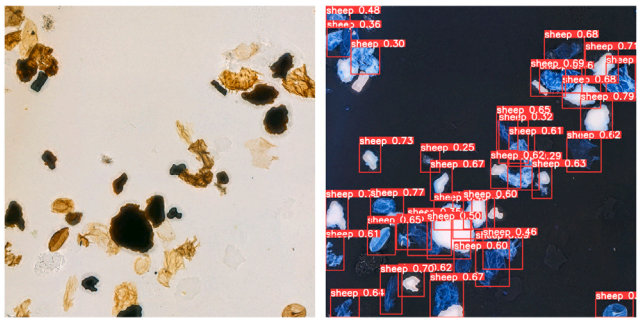


Fig. 4. YOLO object detector. Crop extraction using a pretrained object detection algorithm. Left: Original tile. Right: Resulting bounding boxes. (Admittedly, with a rather sheepish annotation.)

smoothing followed by channel merging, followed by adaptive thresholding using a Gaussian kernel. The thresholded pixels are used to construct coherent regions, where two pixels are defined to be from the same region if they are 2-connected. Regions with a pixel area under a predetermined threshold are discarded. The bounding box for each region is defined to be the smallest rectangle to enclose the region. An illustration of the pipeline is found in Appendix B.

Machine learning approach. In our machine learning approach we aimed to adapt general object detection models previously trained to locate and classify objects on images that were unrelated to our dataset. We used freely available open-source code. Note that we are only interested in good bounding-box proposals at this stage, extracted in a relatively fast manner. We tested several models to find the best compromise between accuracy, fast processing, and simplicity. The model we adopted was an object detection algorithm that was based on YOLOv5 (Redmon et al., 2016). The model was trained to detect bright objects on a dark background (sheep), and is openly available.[5] Note that a high resolution slide is processed one tile at a time in our adaptation.

In order to make an unprocessed tile (Fig. 4 (left)) compliable to the model, we needed to adapt our data by inverting the color intensities before applying the pretrained YOLO model (Fig. 4 (right)). We adjusted the resolution of the input tile to ensure correspondence between the microfossil object size and the expected size for the YOLO model. This adaptation was only done for the bounding-box proposals, and the crops were exported with their original RGB values and the desired resolution.

Due to varying shapes and sizes of the fossils, the crop resolution naturally varies from fossil to fossil. As a deep learning model in general requires that all inputs are of equal shape, all the crops were resized to the same square image resolution. The aspect ratio was not preserved in this step. We used both 96-by-96 and 224-by-224 as target resolutions for the crops to allow for model differences in training. The 96-by-96 resolution crops were used for lightweight training of the CNN, while we used the 224-by-224 resolution for ViT training as this is the standard resolution in ViTs.

Comparison. Fig. 5 shows a comparison of the two approaches. As the slides do not contain any bounding boxes, the comparison is purely qualitative. Both methods work satisfactory for a large part of the data, however, the figure shows that the pretrained YOLO model did a better job overall, particularly at detecting overlapping objects. When utilizing a GPU, the YOLO model were also much faster with a processing time of approximately 5 min per slide compared to 30 min for the image processing method. Other methods such as the Segment Anything Model (SAM) (Kirillov et al., 2023) were too slow for our purposes.

[5] https://huggingface.co/keremberke/yolov5m-aerial-sheep.

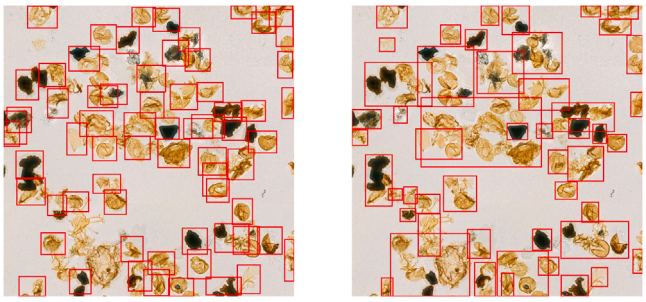


Fig. 5. Comparison of methods. Comparison of methods. Left: Machine learning approach. Right: Pipeline of standard image processing methods. More fossils are separated with the machine learning approach.

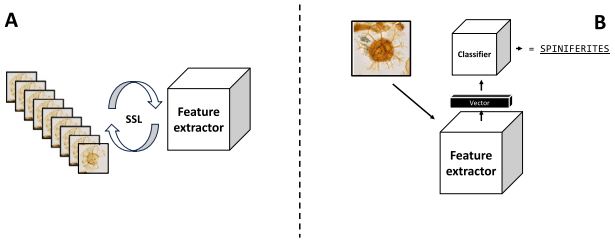


Fig. 6. A: A model is trained self-supervised to extract useful features from fossils. B: The trained model is used to extract features from fossils. The extracted features are used as inputs to a classifier. Only the classifier is trained at this stage.

When employing the machine learning approach, our exact count were 3,386,209 non-overlapping crops for the 215 slides of well NO 6407-6-5. This count includes all detected objects (i.e. palynomorphs and phytoclasts), except for objects that were part of clusters and thus were difficult to separate and isolate.

2.3. Microfossil identification

Our end goal is to train a model to classify microfossils, which is a non-trivial task. Conventional supervised training using image-to-label mapping is not feasible for our problem, as the number of labeled fossils in the data is limited. In addition, our data contains a vast amount of unlabeled data which supervised methods are unable to utilize. Therefore, we propose a two-step approach where, in the first step, we train a self-supervised model which serves as a foundation for any downstream task of interest, for instance classification or clustering of fossils (Fig. 6A). The self-supervised model is trained on unlabeled data with the underlying assumption that self-supervised training will teach the model how to extract relevant features from fossils. When applied in classification (Fig. 6B), the self-supervised model serves as a feature extractor which transforms an image (e.g. with resolution $(224, 224, 3)$) to a feature vector of a much smaller dimension (e.g. with length 376). After completing the self-supervised training, the self-supervised model is applied to the labeled data, where the extracted features are used to train a classifier in a supervised manner, utilizing the fact that the feature vectors now contain dense information about the image.

For the self-supervised model training we focus on SimCLR (Chen et al., 2020a) and DINO (Caron et al., 2021), which are two augmentation-based SSL methods. SSL using augmentations is a paradigm in self-supervised learning where synthetic input variations are used to construct useful learning objectives. For images, this translates into creating two different views of the same input by applying image transformation. The SSL objective is to force the network to create the same feature vector for both views (map both views to the same representation) by using a suitable loss function.

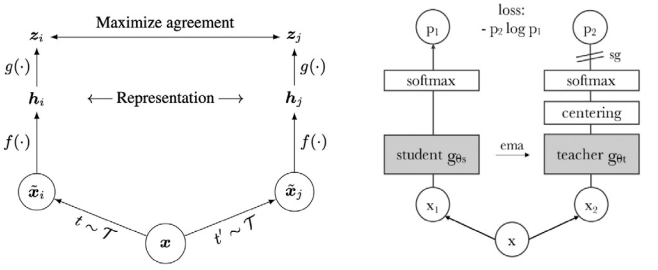


Fig. 7. Outline of both frameworks used in our training. In both frameworks, two views of the same input are mapped to the same latent space representation. Left (SimCLR): SimCLR training outline. *Image taken from Chen et al. (2020a), license CC-BY 4.0.* Right (DINO): DINO training outline. The purpose of the centering stage is to stabilize training (see Caron et al. (2021) for details). *Image taken from* Caron et al. (2021), *license CC-BY 4.0.*

2.3.1. SSL using SimCLR

A frequently applied method in SSL is SimCLR (Chen et al., 2020a), a framework for contrastive learning for CNNs that has achieved excellent results on SSL classification benchmarks. An outline of the framework is illustrated in Fig. 7 (left). SimCLR consists of several operations, where the first step of the pipeline is to convert a batch of input images $\mathbf{x}$ into two variations (or views) $\tilde{\mathbf{x}}_i, \tilde{\mathbf{x}}_j$ using a set of chosen image transformations (a data augmentation pipeline). The two views are then processed by a CNN $f(\cdot)$ to produce latent space representations $(\mathbf{h}_i, \mathbf{h}_j)$ for the views. Finally, the two latent representations are projected onto a lower-dimensional space by a projection head $g(\cdot)$. Note, that while we are primarily interested in the latent space representations, the loss is calculated on the two lower-dimensional representations $(\mathbf{z}_i, \mathbf{z}_j)$ to maximize the similarity between the two views.

The loss function used in SimCLR is called the normalized and temperature-scaled cross entropy (NT-Xent) loss function and is closely related to the commonly used cross-entropy loss function, though slightly more involved. The NT-Xent loss $l_{i,j}$ between the 2 variations $\tilde{\mathbf{x}}_i, \tilde{\mathbf{x}}_j$ is given in the following equation:

$$l_{i,j} = -\log \frac{\exp(\text{sim}(\mathbf{z}_i, \mathbf{z}_j)/\tau)}{\sum_{k=1}^{2N} I\{k \neq i\} \exp(\text{sim}(\mathbf{z}_i, \mathbf{z}_k)/\tau)}, \tag{1}$$

where sim is the cosine similarity, τ is a chosen sharpening parameter to improve training (see Chen et al. (2020a)), N is the batch size, and $I\{\cdot\}$ is an indicator function equal to 1 if $k \neq i$. It penalizes a low cosine similarity between feature vectors originating from the same input image. The loss is only computed for positive pairs, and can be interpreted as a classification task with $2N - 1$ possibilities (N being the batch size). For this reason, the batch size is an important hyperparameter which should be set high (Chen et al., 2020a).

We apply our own Python implementation of SimCLR in our work, utilizing the TensorFlow library (Abadi et al., 2015). As shown previously, the transformations applied to create the two views are critical to produce good results (Chen et al., 2020a). Our data augmentation pipeline consists of translation, zoom, color jittering, and blurring (see Appendix C for details). We use the Xception (Chollet, 2017) without the final classification layer as our CNN, which is an efficient model that utilizes depthwise separable convolutions with a moderate number of parameters. We also apply a MLP projection head consisting of a fully connected layer with 128 units followed by ReLU activations followed by a linear fully connected layer with 128 output units. We used the Adam optimizer (Kingma and Ba, 2017) with an inverse time decay learning rate schedule with an initial learning rate of 0.001 and a decay rate of 0.05. We also used a temperature parameter of 0.1 as in the original paper (Chen et al., 2020a). The training length was set to 100 epochs. The batch size was limited by GPU memory and was set to 128. We used a resolution of 96 by 96 when training the model to increase training efficiency. The source code is available at https://github.com/IverMartinsen/simclr.

2.3.2. SSL using DINO

DINO (Caron et al., 2021) is a self-supervised learning framework that has emerged as a ViT counterpart to CNN-based frameworks. In DINO (Fig. 7 (right)), two views x_1, x_2 of the same input are created by a data augmentation pipeline, before being processed and aligned to the same point in a latent space. The two views are processed by a student and a teacher network $(g_{\theta_s}, g_{\theta_t})$, which are two networks of similar architecture that produce two latent representations (not shown in the figure). The latent representation is further processed by a projection head (sub-model of the encoder, not shown in figure) to create two K-dimensional output vectors p_1, p_2. The output vectors are scaled by a temperature parameter τ, and normalized by the softmax operator. The following equation gives the output P_s of the Student ith component as a probability:

$$P_s(x)^{(i)} = \frac{\exp(g_{\theta_s}(x)^{(i)}/\tau_s)}{\sum_{k=1}^{K} \exp(g_{\theta_s}(x)^{(k)}/\tau_s)}, \tag{2}$$

given the input x. A similar calculation is done for the teacher (t) output. Note that as both the student and the teacher output vectors sum to one, the final outputs can be interpreted as representing parameters of two categorical distributions where the uniformity is controlled by the temperature parameter. The loss is computed as the cross-entropy of p_1 with respect to p_2:

$$\text{Loss}(X, \theta) = \sum_{x \in X} H(P_t(x), P_s(x)), \tag{3}$$

where X is the set of all images, and thus encourages the student to produce outputs (p_1) to match the teacher (p_2). The cross entropy between two categorical distributions p and q is defined as follows:

$$H(p, q) = -\sum_{i=1}^{K} p(i) \log q(i). \tag{4}$$

It is important to note that during training, only the student weights are optimized by gradient descent. The teacher weights are considered constant in the loss calculation and are instead updated using an exponential moving average (*ema*, Fig. 7 (right)) of the updated student weights and previous teacher weights. This is similar to the approach in BYOL (Grill et al., 2020) and SimSiam (Chen and He, 2020), and helps prevent collapsing solutions.

We use the official DINO implementation in our work. Thus, we train our ViTs for 100 epochs using the default hyperparameters when training ViTs from scratch on our data (see Caron et al. (2021) for details). In addition to training ViTs from scratch, we also employ fine-tuning of pretrained ViTs. When fine-tuning, the teacher and student weights are initiated with the pre-trained ImageNet weights obtained in Caron et al. (2021), where the model was trained in a self-supervised fashion on the ImageNet dataset with 1000 classes containing 1,281,167 training images. Our fine-tuned model is trained for 10 epochs with default hyperparameters. We use an input resolution of 224-by-224 as in the original DINO implementation when training our models.

2.3.3. Classifying labeled microfossils

After finalizing SSL model training, it is common to use a simple model such as a logistic regression model or a k nearest neighbors classifier for the final classification (e.g. Chen et al., 2020a; Grill et al., 2020; Chen and He, 2020; Caron et al., 2021). We evaluate our self-supervised models using both a logistic regression model and a k nearest neighbors (kNN) classifier. These models are easy to implement and should perform well if the self-supervised model does a good job in the feature extraction. Other possible classifiers include a single layer perceptron and support vector machines (SVMs).

Due to the class imbalance caused by very few labels for the majority of the genera in the labeled dataset, we decided to select only the 20 most abundant genera for classification, resulting in 20 classes of 697 crops in total. Table 1 shows the final classes. The resulting data set was

Table 2
Results Performance metrics for our trained models compared to pre-trained benchmarks. We tested 3 different models, Xception which is a convolutional neural network, ViT-T the "tiny" version and ViT-S the "small" version of the ViT Vision Transformer model. With SimCLR, we are able to quickly train an encoder that is better than the ImageNet model that is trained supervised on ∼1M images of resolution 224 × 224. The ViT-S ImageNet benchmark is better than the supervised Xception model, and the model is further improved when fine-tuned on our data. The *k*NN classifiers were trained for a range of values for *k*. Results corresponding to the optimal *k* are reported in the upper half, while results for all tested values are reported in the lower half of the table. All evaluation was done with a resolution of 224-by-224 on the labeled data.

	Framework	Architecture	Res.	Param.	Data	*k* nearest neighbors				Logistic regression			
						k	Av.R	Av.P	Acc.	Av.R	Av.P	Acc.	log loss
Benchmark	Supervised	Xception	224	21M	ImageNet	9	0.56	0.64	0.60	0.72	0.76	0.74	0.78
Trained	SimCLR	Xception	96	21M	Microfossils	5	0.65	0.70	0.69	0.79	0.80	0.80	0.70
Trained	DINO	ViT-T	224	6M	Microfossils	5	0.60	0.64	0.63	0.72	0.73	0.73	0.91
Trained	DINO	ViT-S	224	22M	Microfossils	5	0.62	0.67	0.65	0.75	0.76	0.76	0.83
Benchmark	DINO	ViT-S	224	22M	ImageNet	9	0.82	0.84	0.83	0.88	0.89	0.89	0.41
Fine-tuned	DINO	ViT-S	224	22M	Microfossils	5	**0.84**	**0.86**	**0.84**	**0.89**	**0.91**	**0.91**	**0.38**

1 nearest neighbors			3 nearest neighbors			5 nearest neighbors			7 nearest neighbors			9 nearest neighbors		
Av.R	Av.P	Acc.	Av.R	Av.P	Acc.	Av.R	Av.P	Acc.	Av.R	Av.P	Acc.	Av.R	Av.P	Acc.
0.55	0.59	0.58	0.52	0.58	0.56	0.55	0.61	0.59	0.56	0.63	0.60	0.56	0.64	0.60
0.64	0.67	0.67	0.65	0.70	0.68	0.65	0.70	0.69	0.64	0.70	0.68	0.63	0.69	0.67
0.59	0.63	0.63	0.58	0.62	0.61	0.60	0.64	0.63	0.58	0.62	0.61	0.58	0.62	0.62
0.61	0.64	0.63	0.63	0.67	0.64	0.62	0.67	0.65	0.61	0.66	0.64	0.60	0.64	0.63
0.78	0.81	0.80	**0.82**	**0.85**	**0.83**	0.81	0.84	0.82	**0.82**	**0.85**	**0.83**	**0.82**	0.84	**0.83**
0.82	**0.84**	**0.83**	0.81	0.84	0.82	**0.84**	**0.86**	**0.84**	**0.82**	**0.85**	**0.83**	0.81	**0.85**	**0.83**

further split into a training set (80%, 557 images) and a test set (20%, 140 images) using stratified sampling to ensure that the frequency of occurrence was approximately the same for all classes for both sets. This resulted in a class-wise sample size in the range of 4 to 14 for the test set. All classifiers were fitted to the same training partition of the labeled subset (80% of images) and evaluated on the test partition (remaining 20%). The logistic classifier was numerically optimized using a weighting of samples to account for the class imbalance, while the *k*NN classifier was fitted using 1, 3, 5, 7 and 9 neighbors.

To cross-validate the classification results, the evaluation in the last paragraph was repeated 10 times with 10 different train/test splits that were sampled randomly using different seeds. We report the average score across the 10 runs.

2.3.4. Related methods

We also considered other SSL frameworks, in particular BYOL (Grill et al., 2020) and SimSiam (Chen and He, 2020). Preliminary results using our own implementations of BYOL and SimSiam showed worse performances compared to SimCLR.

2.3.5. Data used for training

To reduce resource requirements when training and comparing multiple models, we only used a subset of the available data for training. The data used for training consists of 100,000 crops taken from 22 different slides (selected arbitrarily from the 215 slides). We discarded crops with overlapping bounding boxes to ensure a high-quality training data set without overlapping objects. The 100 K crops used for training are published and openly available (Martinsen et al., 2024).

Although poor generalization and memorization can also be an issue in self-supervised training (Meehan et al., 2023), the 22 selected slides used in the self-supervised training contained only 28 labeled fossils in total. These fossils may or may not be part of the self-supervised training but contribute too small a fraction to have an impact on the test results.

3. Results

Table 2 shows metrics that summarize the performances of all our trained models and benchmark models. In addition to accuracy (Acc.), we report the average recall (Av.R) and precision (Av.P) for both *k*NN and the logistic classifiers. The average recall and precision are computed using the arithmetic mean over all classes and are not affected by class imbalance. These are useful metrics, as the accuracy alone might be misleading when reporting results for imbalanced datasets. We also report the negative log-likelihood (log-loss) for the logistic regression model (not relevant for *k*NN).

Description of models We use three different architectures in our experiments; a single CNN and two different ViTs. The CNN we use is identical to the Xception architecture described in Chollet (2017), and contains approximately 21 million trainable parameters. We compare the Xception model trained self-supervised using SimCLR on our data against the Xception model pretrained supervised on ImageNet.[6]

The two ViT architectures follow the design described in Dosovitskiy et al. (2021) and Caron et al. (2021). Both ViT-T (tiny) and ViT-S (small) has a depth of 6 transformer layers. The two models differ in the embedding dimension (192 for ViT-T and 384 for ViT-S) and the number of attention heads (3 for ViT-T and 6 for ViT-S), resulting in approximately 6 and 22 million trainable parameters, respectively. We compare using four different weights for the ViTs: ViT-S pretrained self-supervised on ImageNet (see Caron et al. (2021)), ViT-T and ViT-S trained self-supervised on our data, and ViT-S fine-tuned on our data. The scores for ViT-S fine-tuned are reported for the best model, which was trained for 6 epochs.[7]

Discussion Comparing CNNs only, Table 2 shows that the best model is obtained using self-supervised training on our data. The SimCLR model outperforms the ImageNet benchmark in all metrics, especially when comparing the metrics for the *k*NN classifiers. For both models, the classification is clearly better with a logistic classifier compared to *k*NN, however the difference in performance between the two classifiers is smaller for the SimCLR model compared to the ImageNet model.

When comparing ViTs only, the best performing model is the ViT-S pretrained on ImageNet and fine-tuned on our data. This model performs slightly better than the ImageNet benchmark model on all metrics and performs much better than the ViT-S and ViT-T models, which were trained from scratch using our data only. The ViT-S is larger than the ViT-T, and as expected performs better since self-supervised training is known to benefit from larger models (Chen et al., 2020a,b; Caron et al., 2021). Note that both the ViTs that were trained

[6] The Xception model was trained supervised on the ImageNet dataset with 1000 classes (Deng et al., 2009) containing 1,281,167 training images.

[7] We fine-tuned the ViT-S for 10 epochs and evaluated the model after each epoch. Although the best model was obtained after 6 epochs, the evaluations for all the 10 training steps showed an improvement compared to the pretrained model.

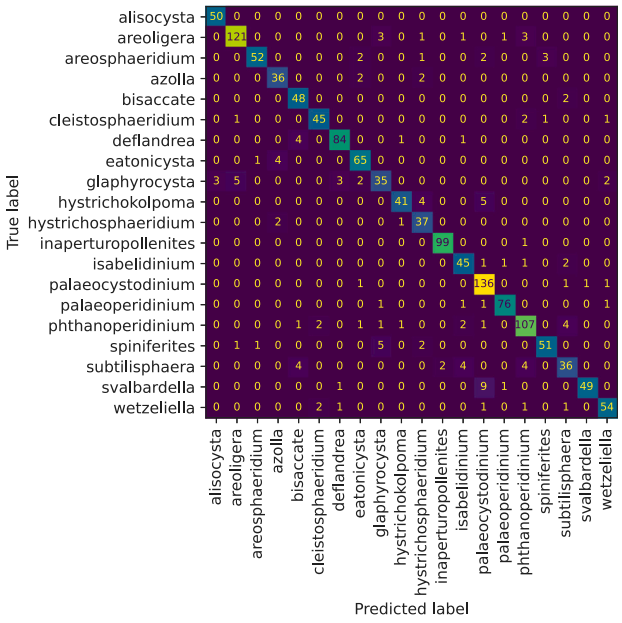


Fig. 8. Confusion matrix The matrix entries are the number of microfossils crops (labeled samples) in the test set. Our best model was obtained with a logistic classifier trained on top of a fine-tuned ViT-S. On average 13 out of 140 test examples were misclassified across each fold (in total 133 out 1400 test examples).

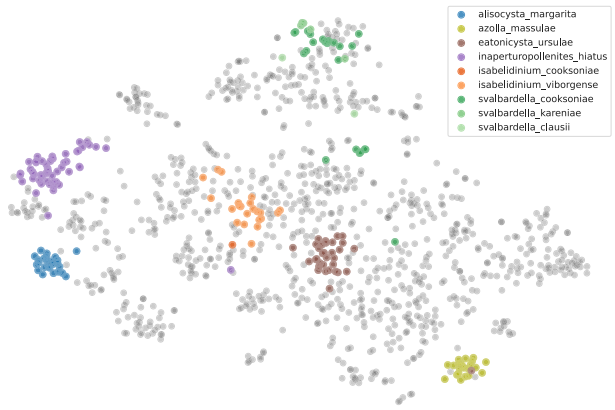


Fig. 9. t-SNE Visualization: the figure shows a projection of the latent space into a 2-dimensional representation for our best performing model. A selection of species is highlighted with color in this figure. The t-SNE representation shows that (1) crops from the same genera are grouped together in the embedding space, (2) species from the same genus (Isabelidinium in different shades of orange and Svalbardella in different shades of green) are closer.

from scratch performed worse than the SimCLR model. ViTs lack the inductive bias of CNNs and are thus known to demand more data in training (Dosovitskiy et al., 2021). We believe that by increasing our dataset for training along with increasing the training time, we would be able to train a ViT from scratch with a substantial increase in performance, perhaps on par with or better than the ImageNet model. Models trained self-supervised are known to benefit from larger training sets, bigger architectures and longer training time with little to no risk of overfitting (Meehan et al., 2023; Chen et al., 2020a; Caron et al., 2021; Grill et al., 2020).

Table 2 also show that the ViT-S ImageNet baseline performs better than the SimCLR model, even if the ViT is trained on completely different data. We do believe, however, that with more training, a bigger model and more data we would significantly improve the performance of our SimCLR model.

Training time All models were trained on a single GPU.[8] Training the Xception model using SimCLR for 100 epochs with 100 K images with a resolution of 96 × 96 resulted in a training time of 5-10 h. The ViT-S was fine-tuned for 10 epochs with a similar training duration. Note that training a ViT from scratch for 100 epochs is much more resource intensive, and the training duration was approximately 50 h.

Analyzing the best model Fig. 8 shows the confusion matrix for the best performing model (ViT-S fine-tuned). The model performs well for the majority of the classes, with 100% precision and recall for class 2 (*Areosphaeridium*), 4 (*Bisaccate*), 5 (*Cleistosphaeridium*), 6 (*Deflandrea*), 10 (*Hystrichosphaeridium*), 11 (*Inaperturopollenites*), 14 (*Palaeoperidinium*), 16 (*Spiniferites*), 18 (*Svalbardella*) and 19 (*Wetzeliella*). Class 17 (*Subtilisphaera*) had the lowest F1 score[9] (0.8) and was confused with class 15 (*Spiniferites*) on two occasions. Using t-distributed stochastic neighbor embedding (t-SNE), we are able to visualize how crops from different species and genera are distributed in the embedding space (Fig. 9). *Alisocysta, Azolla, Eatonicysta, Inaperturopollenites* and *Isabelidinium* form clusters around small neighborhoods, and achieves recall scores of 100%, 90%, 93%, 99% and 90%, respectively (computed from Fig. 8). The three species from the *Svalbardella* genus are scattered to a larger extent, which is reflected by a lower recall score of 82%. For clarity, only six classes (6 genera covering 9 species) are shown in the plot. This provides evidence that the self-supervised model has a similar representation in its latent space for images of the same classes.

4. Conclusion

Our work provides an efficient pipeline for extracting microfossils from whole slide images by combining image processing techniques and deep learning. By utilizing state-of-the-art deep learning methods, we are able to efficiently train an encoder that extracts features useful for other tasks such as identification, grouping and counting of microfossils. We show that our approach needs only a reasonable amount of compute resources for both training and inference. The results we obtain on a labeled dataset show that self-supervised training of deep learning models on microfossils results in a significant improvement compared to existing benchmark models. This approach can be generalized to other applications where the task is to extract millions of patterns or shapes from large images and identify them, such as foraminifera and other microfossils from the geological record.

Funding

This document is the result of a research project funded by SFI Visual Intelligence. Visual Intelligence projects are financially supported by the Research Council of Norway, through its Centre for Research-based Innovation funding scheme (grant no. 309439), and Consortium Partners.

CRediT authorship contribution statement

Iver Martinsen: Writing – review & editing, Writing – original draft, Visualization, Methodology, Investigation, Formal analysis, Data curation, Conceptualization. **David Wade:** Writing – review & editing, Validation, Supervision, Data curation, Conceptualization. **Benjamin Ricaud:** Writing – review & editing, Validation, Supervision, Methodology, Conceptualization. **Fred Godtliebsen:** Writing – review & editing, Validation, Supervision, Project administration.

[8] GeForce RTX 3090 24 GB RAM or similar.

[9] The F1 score is computed as the harmonic mean of precision and recall.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgments

The authors wish to thank Robert Williams at the Norwegian Offshore Directorate (NOD) for archiving and digitalizing the slides that were used in this work. We also thank Equinor for permission to use the label data on which this work depends.

Appendix A. Labeled data

See Fig. 10.

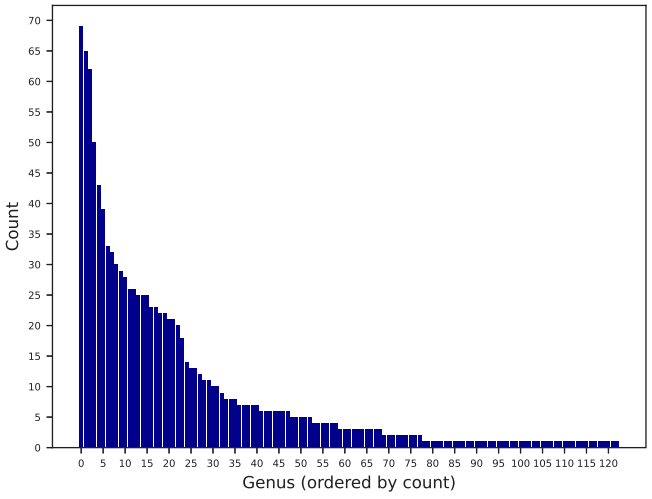


Fig. 10. Bar chart of genera counts. The genera are ordered by abundance. Genus 0 corresponds to the genus with the most labeled examples, while genus 121 corresponds to the genus with the least.

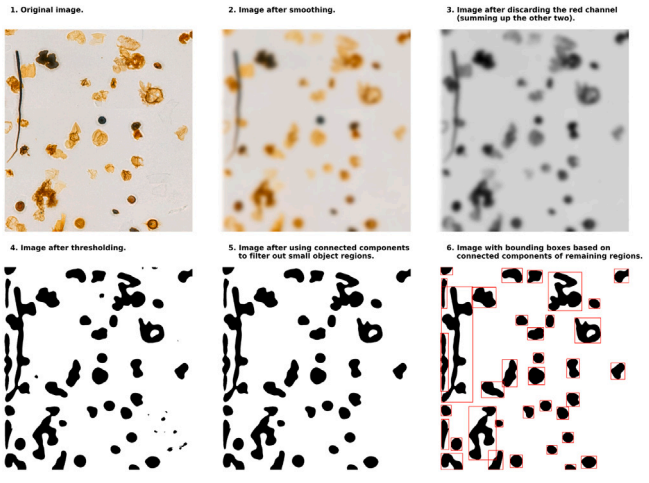


Fig. 11. Obtaining bounding boxes using image analysis. A pipeline for obtaining bounding boxes using classical image analysis methods. A: Original image. B: Image after applying Gaussian smoothing to even out background color. C. Image after discarding the red channel and merging the others. D. Image converted to binary format by thresholding the intensity values. E. Image after using connected components to turn connected regions into objects. F. Image with bounding boxes enclosing the objects.

Appendix B. Extracting microfossils using image processing methods

Fig. 11 shows an outline and a description of how to automatically extract crops from an image by applying classical image processing techniques.

Appendix C. SimCLR implementation details

The following set of augmentations was used in our SimCLR implementation:

- Random horizontal flip.
- Random translation with lower/upper bound of ±25% both horizontally and vertically.
- Random zoom with an upper bound of 50%. The zoom factor is drawn independently for each axis, and as a result the aspect ratio is not preserved.
- Random scaling of pixels with a factor between 0.4 and 0.6 (random brightness adjustment).
- Random color jitter drawn from a uniform distribution with range ±0.2. Jitter is added to the pixel values.
- Smoothing/blurring with a probability of 0.5 using a Gaussian kernel with kernel size of 9 pixels (10% of height/width). The sigma parameter is drawn uniformly from the range $[0.1, 2.0]$.

Preliminary analysis showed that Gaussian blurring and color drop (not used in our training) did not have a big effect on model performance.

References

Abadi, M., Agarwal, A., Barham, P., Brevdo, E., Chen, Z., Citro, C., Corrado, G.S., Davis, A., Dean, J., Devin, M., Ghemawat, S., Goodfellow, I., Harp, A., Irving, G., Isard, M., Jia, Y., Jozefowicz, R., Kaiser, L., Kudlur, M., Levenberg, J., Mané, D., Monga, R., Moore, S., Murray, D., Olah, C., Schuster, M., Shlens, J., Steiner, B., Sutskever, I., Talwar, K., Tucker, P., Vanhoucke, V., Vasudevan, V., Viégas, F., Vinyals, O., Warden, P., Wattenberg, M., Wicke, M., Yu, Y., Zheng, X., 2015. TensorFlow: Large-scale machine learning on heterogeneous systems. URL: https://www.tensorflow.org/. Software available from tensorflow.org.

Baville, P., Apel, M., Hoth, S., Knaust, D., Antoine, C., Carpentier, C., Caumon, G., 2021. Computer-assisted stochastic multi-well correlation: Sedimentary facies versus well distality. Mar. Pet. Geol. 135, 105371. http://dx.doi.org/10.1016/j.marpetgeo.2021.105371.

Caron, M., Touvron, H., Misra, I., Jégou, H., Mairal, J., Bojanowski, P., Joulin, A., 2021. Emerging properties in self-supervised vision transformers. In: International Conference on Computer Vision. pp. 9650–9660, URL: https://openaccess.thecvf.com/content/ICCV2021/html/Caron_Emerging_Properties_in_Self-Supervised_Vision_Transformers_ICCV_2021_paper.html.

Chen, X., He, K., 2020. Exploring simple siamese representation learning. http://dx.doi.org/10.48550/arXiv.2011.10566, URL: http://arxiv.org/abs/2011.10566. arXiv:2011.10566 [cs].

Chen, T., Kornblith, S., Norouzi, M., Hinton, G., 2020a. A simple framework for contrastive learning of visual representations. In: Proceedings of the 37th International Conference on Machine Learning. PMLR, pp. 1597–1607, URL: https://proceedings.mlr.press/v119/chen20j.html. ISSN: 2640-3498.

Chen, T., Kornblith, S., Swersky, K., Norouzi, M., Hinton, G., 2020b. Big self-supervised models are strong semi-supervised learners. http://dx.doi.org/10.48550/arXiv.2006.10029, URL: http://arxiv.org/abs/2006.10029. arXiv:2006.10029 [cs, stat].

Chollet, F., 2017. Xception: Deep learning with depthwise separable convolutions. In: 2017 IEEE Conference on Computer Vision and Pattern Recognition. CVPR, pp. 1251–1258, URL: https://openaccess.thecvf.com/content_cvpr_2017/html/Chollet_Xception_Deep_Learning_CVPR_2017_paper.html.

Deng, J., Dong, W., Socher, R., Li, L.-J., Li, K., Fei-Fei, L., 2009. Imagenet: A large-scale hierarchical image database. In: 2009 IEEE Conference on Computer Vision and Pattern Recognition. IEEE, pp. 248–255.

Dosovitskiy, A., Beyer, L., Kolesnikov, A., Weissenborn, D., Zhai, X., Unterthiner, T., Dehghani, M., Minderer, M., Heigold, G., Gelly, S., Uszkoreit, J., Houlsby, N., 2021. An image is worth 16x16 words: Transformers for image recognition at scale. http://dx.doi.org/10.48550/arXiv.2010.11929, URL: http://arxiv.org/abs/2010.11929. arXiv:2010.11929 [cs].

Goodfellow, I., Bengio, Y., Courville, A., 2016. Deep Learning. MIT Press, http://www.deeplearningbook.org.

Grill, J.-B., Strub, F., Altché, F., Tallec, C., Richemond, P.H., Buchatskaya, E., Doersch, C., Pires, B.A., Guo, Z.D., Azar, M.G., Piot, B., Kavukcuoglu, K., Munos, R., Valko, M., 2020. Bootstrap your own latent: A new approach to self-supervised learning. http://dx.doi.org/10.48550/arXiv.2006.07733, URL: http://arxiv.org/abs/2006.07733. arXiv:2006.07733 [cs, stat].

Guo, J., Han, K., Wu, H., Tang, Y., Chen, X., Wang, Y., Xu, C., 2022. CMT: Convolutional neural networks meet vision transformers. pp. 12175–12185, URL: https://openaccess.thecvf.com/content/CVPR2022/html/Guo_CMT_Convolutional_Neural_Networks_Meet_Vision_Transformers_CVPR_2022_paper.html.

Halbritter, H., Ulrich, S., Grímsson, F., Weber, M., Zetter, R., Hesse, M., Buchner, R., Svojtka, M., Frosch-Radivo, A., 2018. Methods in Palynology. In: Halbritter, H., Ulrich, S., Grímsson, F., Weber, M., Zetter, R., Hesse, M., Buchner, R., Svojtka, M., Frosch-Radivo, A. (Eds.), Illustrated Pollen Terminology. Springer International Publishing, Cham, pp. 97–127. http://dx.doi.org/10.1007/978-3-319-71365-6_6.

He, K., Chen, X., Xie, S., Li, Y., Dollár, P., Girshick, R., 2021. Masked autoencoders are scalable vision learners. http://dx.doi.org/10.48550/arXiv.2111.06377, URL: http://arxiv.org/abs/2111.06377. arXiv:2111.06377 [cs].

He, K., Gkioxari, G., Dollár, P., Girshick, R., 2018. Mask R-CNN. http://dx.doi.org/10.48550/arXiv.1703.06870, URL: http://arxiv.org/abs/1703.06870. arXiv:1703.06870 [cs].

He, K., Zhang, X., Ren, S., Sun, J., 2015. Deep residual learning for image recognition. http://dx.doi.org/10.48550/arXiv.1512.03385, URL: http://arxiv.org/abs/1512.03385. arXiv:1512.03385 [cs].

Johansen, T.H., Sørensen, S.A., 2020. Towards detection and classification of microscopic foraminifera using transfer learning. http://dx.doi.org/10.48550/arXiv.2001.04782, URL: http://arxiv.org/abs/2001.04782. arXiv:2001.04782 [cs, stat].

Johansen, T.H., Sørensen, S.A., Møllersen, K., Godtliebsen, F., 2021. Instance segmentation of microscopic foraminifera. Appl. Sci. 11 (14), 6543. http://dx.doi.org/10.3390/app11146543, URL: https://www.mdpi.com/2076-3417/11/14/6543. Number: 14 Publisher: Multidisciplinary Digital Publishing Institute.

Kingma, D.P., Ba, J., 2017. Adam: A method for stochastic optimization. http://dx.doi.org/10.48550/arXiv.1412.6980, URL: http://arxiv.org/abs/1412.6980. arXiv:1412.6980 [cs].

Kirillov, A., Mintun, E., Ravi, N., Mao, H., Rolland, C., Gustafson, L., Xiao, T., Whitehead, S., Berg, A.C., Lo, W.-Y., Dollár, P., Girshick, R., 2023. Segment Anything. http://dx.doi.org/10.48550/arXiv.2304.02643, URL: http://arxiv.org/abs/2304.02643. arXiv:2304.02643 [cs].

Martinsen, I., Ricaud, B., Godtliebsen, F., Wade, D., 2024. Replication data for: The 3-billion fossil question: How to automate classification of microfossils. http://dx.doi.org/10.18710/KWP9WA, URL: https://doi.org/10.18710/KWP9WA.

Meehan, C., Bordes, F., Vincent, P., Chaudhuri, K., Guo, C., 2023. Do SSL models have Déjà Vu? a case of unintended memorization in self-supervised learning. http://dx.doi.org/10.48550/arXiv.2304.13850, URL: http://arxiv.org/abs/2304.13850. arXiv:2304.13850 [cs].

Raghu, M., Unterthiner, T., Kornblith, S., Zhang, C., Dosovitskiy, A., 2021. Do vision transformers see like convolutional neural networks? In: Advances in Neural Information Processing Systems. Vol. 34, Curran Associates, Inc., pp. 12116–12128, URL: https://proceedings.neurips.cc/paper_files/paper/2021/hash/652cf38361a209088302ba2b8b7f51e0-Abstract.html.

Redmon, J., Divvala, S., Girshick, R., Farhadi, A., 2016. You only look once: Unified, real-time object detection. http://dx.doi.org/10.48550/arXiv.1506.02640, URL: http://arxiv.org/abs/1506.02640. arXiv:1506.02640 [cs].

Smith, T., Waterman, M., 1980. New stratigraphic correlation techniques. J. Geol. 88, http://dx.doi.org/10.1086/628528.

Vincent, P., Larochelle, H., Bengio, Y., Manzagol, P.-A., 2008. Extracting and composing robust features with denoising autoencoders. In: Proceedings of the 25th International Conference on Machine Learning. ICML '08, Association for Computing Machinery, New York, NY, USA, pp. 1096–1103. http://dx.doi.org/10.1145/1390156.1390294, URL: https://dl.acm.org/doi/10.1145/1390156.1390294.

Wheeler, H.E., 1958. Time-Stratigraphy1. AAPG Bull. 42 (5), 1047–1063. http://dx.doi.org/10.1306/0BDA5AF2-16BD-11D7-8645000102C1865D.