https://doi.org/10.5194/egusphere-2026-2075
Preprint. Discussion started: 22 April 2026

Technical note: a Recognition-assisted Camera for Automated Microscopy (RaCAM)

Martin Tetard[1] and Ross Marchant[2]

[1]ESNZ, Lower Hutt, New Zealand.
[2]InFarm, Goondiwindi, Australia.

Correspondence: Martin Tetard (martin.tetard@earthsciences.nz)

Abstract. Automated microscopy workflows, including image acquisition, processing, and recognition using artificial intelligence (AI) are getting a growing interest from the scientific community in biogeosciences, as more and more research institutes are actively working on building datasets of images to train artificial convolutional neural networks (CNNs) to identify microscopic objects.

Here, we present a new, affordable, AI-assisted, Raspberry Pi-powered camera, with the first, built-in, and fully automated microscopy workflow (including automated image acquisition, processing and recognition) that can fit any microscope equipped with a regular C-mount (or CS-mount) camera thread. This camera is equipped with an integrated Single-Board Computer (Raspberry Pi 5) and high-resolution camera sensor (12.3 mp), attached together using a 3D-printable adaptor. Using a new open-source software (RaCAM user interface), written using the Python language, and freely downloadable too, the camera is capable of performing automated acquisition of field of view images, segmenting each visible object of interest, and identifying them using trained CNN onnx models in a few seconds as part of a whole automated workflow.

The camera is also adapted to on-field tasks such as core description, biostratigraphy or even palaeoenvironmental reconstructions based on microfossils census data or morphometry, as it can operate without the need for a spare computer and run directly on a power bank. Finally, as the RaCAM workflow relies on images directly captured by the camera, applications can also be extended outside of the microscopy and micropaleontology research fields as any picture acquired with this device can virtually be processed by the automated workflow.

1 Introduction

The past decades, and especially the last few years, have seen the development of artificial intelligence (AI) technology such as deep learning methods, applied to the training of convolutional neural networks (CNNs) to automatically recognise datasets of images. Rapidly, this technology has been used in the field of micropalaeontology to detect and recognise images of microscopic objects. Various groups of microfossils have been investigated so far using deep learning techniques, including coccoliths (Dollfus and Beaufort, 1999; Beaufort and Dollfus, 2004), foraminifera (Mitra et al., 2019; Hsiang et al., 2019; Marchant et al., 2020; Richmond et al., 2022; Govindankutty Menon et al., 2023; Piva et al., 2024), pollen (Daood et al., 2018; Bourel et al., 2020; Gimenez et al., 2024), radiolaria (Itaki et al., 2020; Tetard et al., 2020; Carlsson et al., 2022, 2023; Tetard et

https://doi.org/10.5194/egusphere-2026-2075
Preprint. Discussion started: 22 April 2026

al., 2023; Mimura et al., 2024, 2025), or even diatoms (Zhang et al., 2023). Automated recognition, especially when integrated into a fully automated microscopy workflow, including image acquisition, processing, and recognition, allows for fast analysis and counting of microscopic object images, as well as generation of morphometric data, some of which would not be possible without automated measurement (e.g., shape parameters). These automated workflows allow one to save a substantial amount of time for microfossil counting, as several hundreds of specimens typically have to be counted over hundreds of samples, a task historically performed manually by an operator, also requiring substantial taxonomic knowledge.

This technology, at first requiring a high-performance computer, gradually became more and more accessible to regular machines over the past years. Several initiatives have recently emerged regarding affordable microscopy based on Raspberry Pi boards and camera modules (Raspberry Pi Ltd., Johnston and Cox (2017), that can be easily programmed for any need such as microfluidic applications, environmental monitoring of plankton, or even biological research to cite a few). Some initiatives focused on building a whole Raspberry Pi-assisted microscope from existing (Whittaker, 2020; Lynn, 2019), or 3D-printed parts (Grant et al., 2020; William, 2022, 2023), but few of these are actually key-in-hand solutions, as the software part usually relies on manual acquisition of images from command line terminal and / or manual processing of these images (Webb, 2017; Nowosad, 2025), and none of them integrate an automated microscopy workflow software capable of performing automated recognition relying on trained CNNs. Schneidereit et al. (2017) also focused on building and automating a motorized stage that could be fitted on a microscope to automatically capture field of view (FOVs) for entire microscope slides.

As development on Raspberry Pi boards using Python libraries and machine learning are becoming more and more accessible, and micro-computers and camera sensors cheaper and more powerful over time, automated tools and deep learning can now be directly used on such compact machines. We decided to develop a portable, affordable, and high-resolution solution that can fit most microscopes in laboratories or directly in the field, and can perform a whole automated microscopy workflow including image capture, processing, and labeling using image segmentation and classification techniques: the Recognition-assisted Camera for Automated Microscopy (RaCAM).

Such a low-cost and key-in-hand solution could be attractive for replacing aging camera units, or developing automated microscopy in emerging countries without sacrificing image quality, for example following MIT (Massachusetts Institute of Technology) involvement in low-cost and open-source, yet high-impact technology for science, education, and community development (e.g., MIT D-Lab (Hoffecker, 2018); MIT HEALS Collaborative developing low-cost diagnostics using AI for life sciences research: https://betterworld.mit.edu/mit-heals/; last access: 21/01/2026). Such initiatives highlight a commitment to reducing the economic barriers of high-cost scientific tools by providing "do-it-yourself" solutions to science. It can also be of significant interest for deployment of analytical clusters (e.g., participative scientific project willing to equip a fleet of boats or ARGO floats (Wong et al., 2020); field water quality assessments using freshwater diatoms; scientific expedition where anyone can operate the microscope without the need for several taxonomic experts to be deployed; oil company working on many boreholes at once and needing a system that can handle a massive amount of microfossils and other microscopic objects to facilitate stratigraphic correlations (Martinsen et al., 2026)).

https://doi.org/10.5194/egusphere-2026-2075
Preprint. Discussion started: 22 April 2026

2 Material and methods

Through the following sections, we will present and explain how to build the RaCAM camera, as well as how to install all required packages and software needed to perform automated image acquisition, processing and recognition directly using the RaCAM.

2.1 Hardware

2.1.1 Cameras

We recommend using either the Raspberry Pi HQ camera (https://www.raspberrypi.com/documentation/accessories/camera.html#hq-camera; Sony IMX477 sensor with a built-in CMOS adaptor, compatible with most microscope mounts), AI camera (https://www.raspberrypi.com/documentation/accessories/ai-camera.html; Sony IMX500 sensor, comes with a built-in AI chip), or camera module V2 (https://www.raspberrypi.com/documentation/accessories/camera.html#camera-module-2; Sony IMX219 sensor, cheapest, but lower resolution. We opted for the HQ camera (12.3 mp; physical pixel size of 1.55 × 1.55 μm) as it is delivered with a CMOS camera adapter, and no built-in lens, allowing for direct microscope image acquisition straight-out of the box, without having to apply a lens shading correction to all acquired images when the factory lens is removed from its sensor. In case a camera module V2 or AI camera is used, a lens shading correction recalibration has to be performed to prevent discolouration of the acquired images after removing the factory lens for microscope use. As the AI camera sensor comes with a built-in AI chip, allowing for faster AI inference, it might also be a wise choice to develop and use neural networks through this sensor.

The camera module V3 (https://www.raspberrypi.com/documentation/accessories/camera.html#camera-module-3; Sony IMX708 sensor), is also cheap and equipped with built-in motorised autofocus, but making the removal of the built-in lens a critical step (on top of disabling the motorised autofocus). Using this camera module V3 for our application is thus not recommended.

Table 1. Comparison between Raspberry Pi camera modules specifications.

Camera name	Sensor denomination	Resolution	Sensor resolution	Pixel size	Price (USD)	Autofocus	AI
Camera module V2	Sony IMX219	8 mp	3280 × 2464 px	1.12 × 1.12 μm	25	No	No
Camera module V3	Sony IMX708	11.9 mp	4608 × 2592 px	1.4 × 1.4 μm	25	Yes	No
HQ camera	Sony IMX477	12.3 mp	4056 × 3040 px	1.55 × 1.55 μm	50	No	No
AI camera	Sony IMX500	12.3 mp	4056 × 3040 px	1.55 × 1.55 μm	70	No	Yes
Global shutter camera	Sony IMX296	1.58 mp	1456 × 1088 px	3.45 × 3.45 μm	50	No	No

https://doi.org/10.5194/egusphere-2026-2075
Preprint. Discussion started: 22 April 2026

2.1.2 Board and accessories

To achieve the best possible performance in terms of acquisition speed / quality / processing of images, and to prevent compatibility issues between all camera parts, we recommend using at least a Raspberry Pi 5 8 GB board, an official fan module, an official 27 W USB-C power supply (providing up to 3 A at 5 V; or a power bank), and at least a 32 GB high-speed micro-SD card (including A2, U3, V30 specs; mandatory to ensure the board functions at a proper speed).

2.1.3 Costs

Depending on configuration, and specific purposes, one might need to choose a board with more RAM, or a different camera sensor, but the costs should remain low. Including the recommended Compute Module 5 (CM5) 8 GB version board (∼100 EUR; prices vary from ∼50 to ∼150 EUR for the 1 GB to the 16 GB versions), HQ camera module (∼ 55 EUR; prices and specifications for all camera can be found in Table 1), fan (∼5 EUR) and high-speed microSD card (∼10 EUR), the total for a micro-computer built-in RaCAM camera is about ∼170 euros (∼120 to ∼220 EUR depending on the board version).

On top of the RaCAM camera, the cost of an official Raspberry Pi screen (∼120 EUR), micro-HDMI to HDMI cable (∼6 EUR), keyboard (18 EUR), mouse (∼8 EUR), and power supply (∼13 EUR) add another 165 euros, for a total cost of ∼340 euros for the full set-up. All software and packages presented in the next section, needed to run the automated workflow, are free.

2.1.4 Camera adaptors and cases

Using the Fusion360 3D-design software, 3D-printable (preferentially resin), Raspberry Pi 5-compatible board cases and camera module adaptors were sketched, designed (Figs 1 and 2), printed, and are freely accessible as supplementary data and at: https://github.com/microfossil/particle-classification-onnx. These 3D models (RaCAM_3D_files.zip © 2026 by Martin Tetard) are licensed under CC BY-NC-SA 4.0. To view a copy of this license, visit https://creativecommons.org/licenses/by-nc-sa/4.0/. This licence is allowing use, modification, and redistribution under the same terms, for personal purposes only as long as credit is given to the creator, and preventing any commercial use.

The camera module adaptors (for the HQ camera, AI camera module, and camera module V2) enable the camera module to be directly attached to a board case, using a total of four to eight M2 screws and nuts. The AI camera module adaptor (and camera module V2) design includes a C-mount and CS-mount threaded version (actual usability largely depends on the 3D-printer, and on the print quality; resin SLA printing is recommended). A second 3D-printable version of the AI camera module (and camera module V2) adaptor to attach the camera to a common C-mount converter that can be bought on third-party websites is also included in the downloadable package.

2.1.5 Assembling and mounting the camera

After 3D-printing one of the adaptor compatible with the camera module, both can be screwed together and fastened to the board case (Fig. 3c) through the vent holes located under it, using four M2 screws and nuts. The camera module ribbon cable

https://doi.org/10.5194/egusphere-2026-2075
Preprint. Discussion started: 22 April 2026

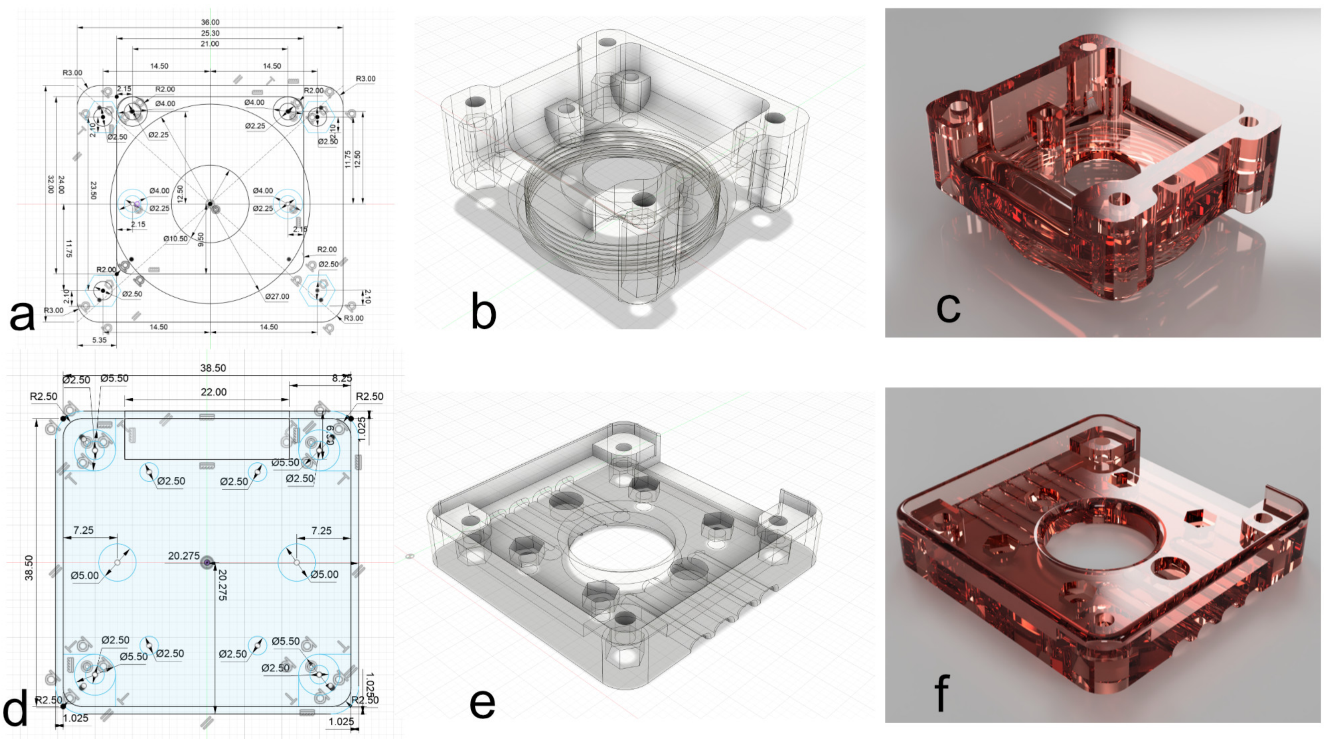

Figure 1. AI camera module (and camera module V2; a, b, c), and HQ camera (d, e, f) adaptors to fix the camera module to the board case. **a**. AI camera module adaptor schematic view. **b**. AI camera module adaptor 3D view.**c**. AI camera module adaptor rendering view. **d**. HQ camera module adaptor schematic view. **e**. HQ camera module adaptor 3D view. **f**. HQ camera module adaptor rendering view.

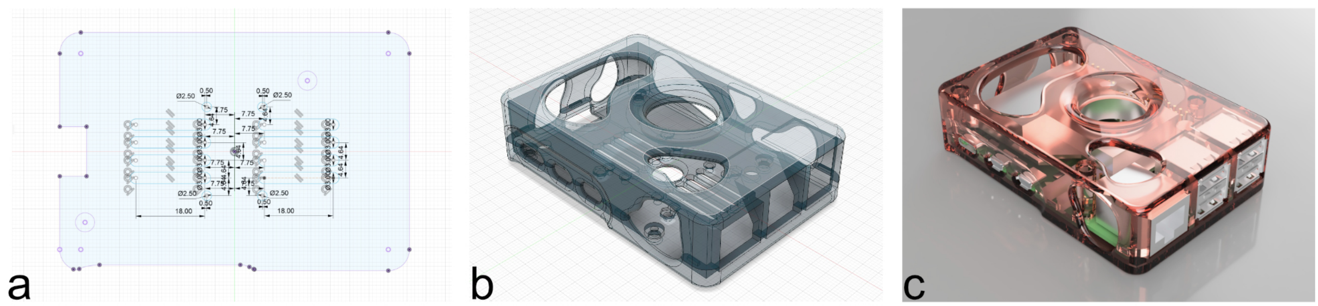

Figure 2. a. Camera board case schematic view. **b**. Camera board case 3D view.**c**. Camera board case rendering view.

can be plugged the "CAM/DISP 0" port on the board, and the board can be secured into its case and closed using M2 screws and nuts.

https://doi.org/10.5194/egusphere-2026-2075
Preprint. Discussion started: 22 April 2026

Next, the camera can be fastened on the C-mount thread usually located on top of the microscope. It can then be connected to a USB-C power supply (or power bank), and connected to a screen using a mini-HDMI to HDMI cable. Finally, a keyboard and mouse can be connected to any USB-A ports.

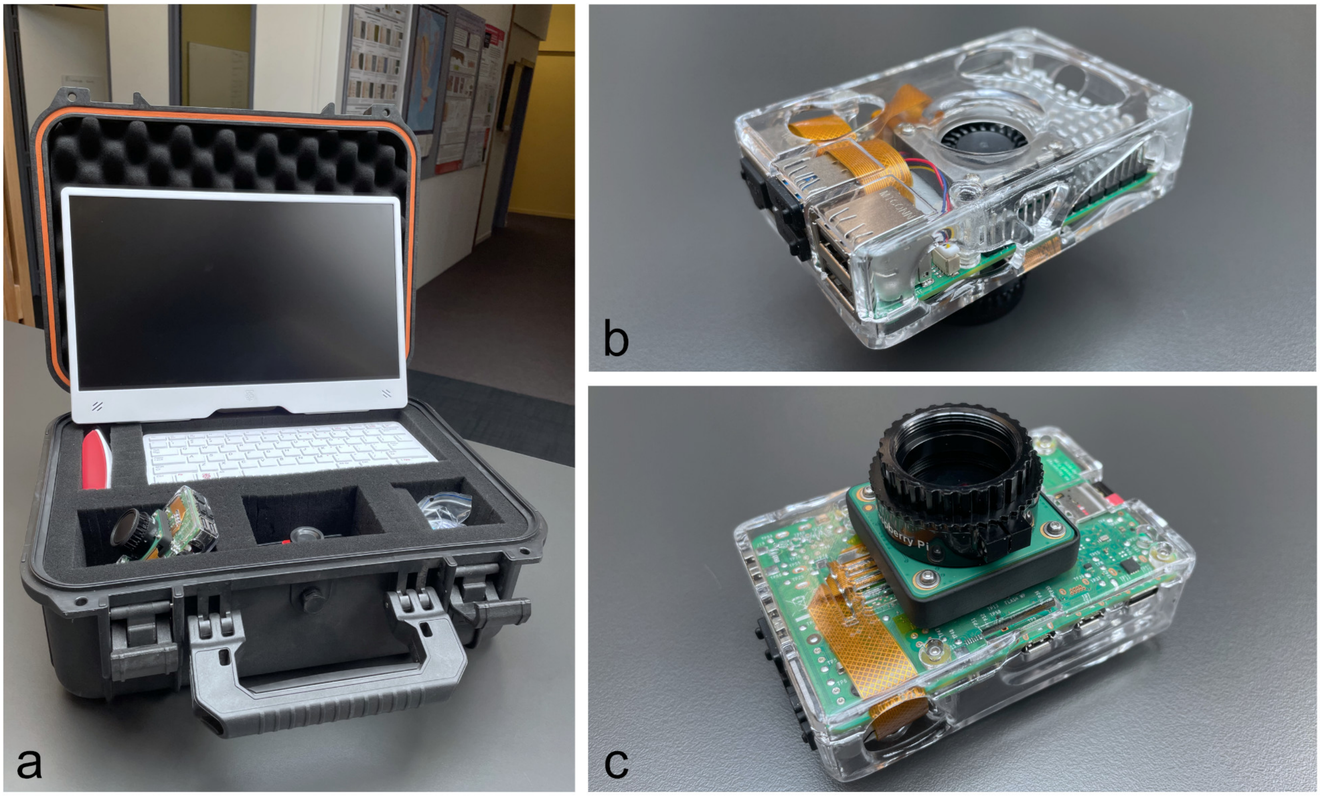

Figure 3. a. RaCAM full set-up with transport and storage hard case. **b**. Upper side view. **c**. Lower side view.

2.2 Software

This section is dedicated to a step-by-step installation procedure of every software and package required to run the whole automated microscopy workflow on the RaCAM camera. This procedure is also detailed in the `Installationprocedure` file included in the `RaCAM_software.zip` file (located in the `RaCAM_files/` directory), available as supplementary data and at: https://github.com/microfossil/particle-classification-onnx.

The `RaCAM_software.zip` (containing software, ImageJ scripts, and miso-onnx library developed in this study), is protected under GNU GPLv.3 license (Copyright © 2026 Martin Tetard). This license allows users to freely run, share, and modify software while requiring that any modified versions be distributed under the same license terms, including the disclosure of source code.

The whole automated workflow can also be operated in "server mode" configuration (image acquisition and processing / segmentation performed on the camera, while recognition is performed on a remote computer). To do so, please also follow the instructions in the `Servermodeprocedure` file and the `RaCAMservermode.py` script.

2.2.1 OS installation

Micro-computers / boards are usually key-in-hand and only require a micro-SD card loaded with the latest version of Raspberry OS (Linux distribution). If not included, a tutorial to easily format and install the last OS on a micro-SD card using the Raspberry Pi Imager software can be found at: https://www.raspberrypi.com/documentation/computers/getting-started.html#raspberry-pi-imager. The current RaCAM software was developed in November 2025 on Raspberry Pi OS (64-bit) running Debian version 13 (Trixie).

2.2.2 Downloading and installing Python packages

Some specific Python packages are required to run the RaCAM software operating the whole automated microscopy workflow. To download and install them, open a terminal window and run:

sudo apt update
sudo apt upgrade
sudo apt install python3-pip
sudo apt install python3-tk
sudo apt install python3-pil
sudo apt install python3-pil.imagetk

2.2.3 Downloading and installing the RaCAM software

The RaCAM software is responsible for controlling the whole automated workflow, from image acquisition (automatically run through the default terminal using automatically generated command lines), image processing and segmentation (automatically run through the ImageJ software), to image recognition (automatically run through an Anaconda terminal and virtual environment using generated command lines). To check if your camera is detected and working properly, open a terminal windows and run:

rpicam-hello --timeout 0

For more information about built-in camera software and rpicam library, the reader is referred to https://www.raspberrypi.com/documentation/computers/camera_software.html.

The RaCAM software package (`RaCAM_software.zip`) can be freely downloaded at: https://github.com/microfossil/particle-classification-onnx (and available as supplementary data). It is an open-source software, using open-source packages and software. To install the RaCAM software:

https://doi.org/10.5194/egusphere-2026-2075
Preprint. Discussion started: 22 April 2026

-Unzip the RaCAM_software.zip file on the camera Desktop and ensure that the RaCAM file, and /RaCAM_files/ and /RaCAM_output/ folders are located on the desktop.

-Open a file manager window, navigate to the "Edit" menu: "Preferences" and select "Don't ask option on launch of executable files" in the "General" tab.

-Open the RaCAM file located on the Desktop with a text editor, and update lines 3 (Exec=) and 4 (Icon=) by replacing <user> with your board username, and save it. You should enter your OS profile username currently in use, which can be found by looking at the path to the Desktop directory (e.g.: /home/<user>/Desktop/). You should now see the RaCAM file icon as the camera case after a restart.

-Double-click on the RaCAM icon to start the RaCAM software, you will be asked to enter your username for updating paths for running the software. Once validated, the software should start successfully and you should be able to see the user interface (Fig. 4).

-The RaCAM software can now be closed as other software and packages are required before using it.

2.2.4 Download and installing ImageJ

ImageJ is an open-source software (Schneider et al., 2012) dedicated to image analysis and image processing. We used it in our workflow to automatically process the original FOVs that are captured during the acquisition step and segment every particle / microfossil into individual object images. To install it on the Raspberry Pi OS:

-Click on the top-left "Application menu bar" icon: "Preferences": "Add / Remove software" and look for "Image processing program with a focus on microscopy images" and "Java library for ImageJ". Select both of them then click "OK".

-Start it in "Application menu bar" under "Education", and close it.

-Navigate to /home/<user>/. Press "Ctrl+H" to show hidden files and folders.

-Go into the /home/<user>/.imagej/ directory.

-Copy the AutoDiato_RaCAMx40.ijm and Adjustable_Watershed.class files (located in /home/<user>/Desktop/RaCAM_files/imagej_plugins/) into the /home/<user>/.imagej/macros/ and /home/<user>/.imagej/plugins/, respectively.

The AutoRadio_RaCAMx10.ijm script, originally published in Tetard et al. (2020) allows for batch processing of original FOVs. It is provided to allow the user to check that the workflow is working correctly by identifying an example set of images of marine plankton specimens (Eocene Radiolarian). It is specialised for segmentation of radiolarian specimens captured using a x10 lens, but parameters can be easily modified for the segmentation of other microscopic objects, acquired at various magnification (and directly from the RaCAM software interface).

2.2.5 Downloading and installing Miniconda3

Miniconda3 is an open-source environment manager that is used to handle specific Python packages and libraries installation through a virtual environment that cannot be used directly in the default OS environment. It is used in our workflow to auto-

https://doi.org/10.5194/egusphere-2026-2075
Preprint. Discussion started: 22 April 2026

matically perform inference of segmented object images using pre-trained onnx models (based on ResNet50 CNNs) and the miso-onnx library (modified after (Marchant et al., 2020)). To install Miniconda3, open a regular terminal window, and run:

wget repo.anaconda.com

wget https://repo.anaconda.com/miniconda/Miniconda3-latest-Linux-aarch64.sh

bash /Miniconda3-latest-Linux-aarch64.sh

Closely follow instructions during Miniconda3 installation. Answer "yes" to the license terms, and install it in the default location (/home/<user>/miniconda3/), and answer "yes" when asked to proceed to initialization. Next, to install packages in Miniconda3 environment, run and accept main terms of service and proceed to the installation:

conda create -n miso-onnx python=3.11

conda activate miso-onnx

pip install opencv-python

pip install numpy

pip install torch

pip install pillow

pip install onnxruntime

pip install git+https://github.com/microfossil/particle-classification-onnx

Finally, check if the "miso-onnx" installation is successful by testing an onnx model inference using a CNN model pre-trained on Eocene radiolarian images (Tetard et al., 2020), and examples of radiolarian specimen images by running (after replacing the three <user> fields with your username):

miso-onnx classify --network-info /home/<user>/Desktop/RaCAM_files/CNN_models/ResNet50_EoceneRadiolaria/model_onnx/network_info.xml --images /home/<user>/Desktop/RaCAM_files/CNN_models/ResNet50_EoceneRadiolaria/simple_test/unlabeled_images --output-csv /home/<user>/Desktop/RaCAM_files/CNN_models/ResNet50_EoceneRadiolaria/simple_test/prediction_file/predictions.csv

This operation should take a few seconds and should generate a .csv file located in /RaCAM_files/CNN_models/ResNet50_EoceneRadiolaria/simple_test/prediction_file/. For more information about the use of onnx models in a recognition workflow, the reader is referred to: https://github.com/microfossil/particle-classification-onnx. For this study, all images are greyscale as in Tetard et al. (2020), and input image size is reduced to 256x256 px^2 for memory saving purposes. The chosen architecture for CNN training of the example CNN provided here for Eocene radiolaria is the ResNet50 with gain and cyclic layers, commonly used for microfossil identification (Mitra et al., 2019; Marchant et al., 2020; Bourel et al., 2020; Tetard et al., 2020; Carlsson et al., 2023; Tetard et al., 2023). This pre-trained model is focused on Eocene radiolaria, but the workflow can be applied to any microfossils observable under a transmitted or stereoscopic microscope. Using the ParticleTrieur software (Marchant et al., 2020), and documentation freely available at https://particle-classification.readthedocs.io/en/latest/installation.html, the user can also easily train their own model, based on their own dataset of images and upload it on the RaCAM camera.

https://doi.org/10.5194/egusphere-2026-2075
Preprint. Discussion started: 22 April 2026

3 Results and Discussion

3.1 Automated microscopy workflow using the RaCAM software

When starting up the RaCAM software, the user is presented with a user-friendly interface (Fig. 4). Through this interface, the user can choose to perform image acquisition, and / or image processing, and / or image recognition. You must choose a saving / working directory, an ImageJ script, and a CNN model to do so, respectively. The user should also fill the "Core_name" and "Sample_name" fields (which can also be used to account for any other kind of metadata if the acquisition does not follow a `/Core/Sample/` structure). The original FOV images, segmented object images, and labeled images will then be automatically saved following a `/Core_name/Sample_name/Label/` path structure (Fig. 5). Images generated during the image acquisition, processing, and recognition steps of the workflow will be saved in the `/RaCAM_output/Image_acquisition/`, `/RaCAM_output/Image_processing/`, and `/RaCAM_output/Image_recognition/` directories, respectively.

The user can also adjust the image capture, processing, and labeling parameters, such as brightness, contrast, exposure time, image (camera) rotation, image quality (% jpeg quality), image type (8-bit greyscale or colour), segmented object image quality, minimum size for an object to be saved (px^2), and finally % certainty for a CNN to label an object image. On the top menu bar, instructions are detailed for each field that needs to be filled. Finally, the four bottom buttons can be used to show a livefeed preview of the field of view before acquisition (should be closed before performing actual image acquisition), capture the image and perform the whole workflow as intended (acquisition; acquisition and processing; or acquisition, processing and recognition), batch process an existing folder of images, or generate a census data file (.csv table), respectively.

3.2 Manually operating the image acquisition, image processing or image recognition

Using the "Batch process" button, the user can also batch process an existing folder of FOV images for segmentation, or do the batch recognition of existing folders of object images as the automated workflow consists of three separated modules that can be operated individually. For this purpose, the user just needs to select "No" for Image acquisition and / or "No" for the image processing on the RaCAM software interface, and click "Batch processing" button after selecting an ImageJ script and CNN model. For the recognition step, object images that will be labeled are those located in `/RaCAM_output/Temp/`.

3.3 Generating census data

The "Census data" button is used to generate a `.csv` file containing census counts for the investigated core. This function will scan the `/Label/` subdirectories of each `/RaCAM_output/Image_recognition/Core_name/Sample_name/` directory and compile the number of images of each subdirectory (taxa abundance) into a three-column table showing the "Sample name", "Label name", and "Label abundance" of each label per sample in the core. When the "Census data" button is clicked, the user will be asked to choose a core directory, which must be selected to successfully perform the census data generation.

https://doi.org/10.5194/egusphere-2026-2075
Preprint. Discussion started: 22 April 2026

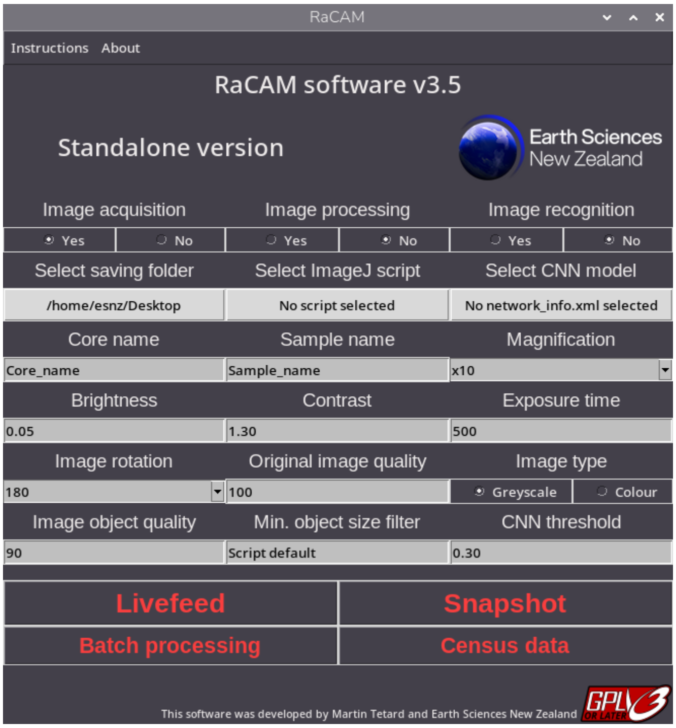

Figure 4. RaCAM software user interface.

3.4 Overall performance

With a physical pixel size of 1.55 μm (for an effective pixel size of about 0.155 to 0.015 μm if used together with x10 to x100 magnification lenses), the RaCAM camera resolution is comparable to an high-end camera for regular (i.e., stereoscopic or transmitted light) microscopy for a fraction of their price, As the camera is also able to directly perform image processing and recognition, and without any additional costs for any proprietary software, it is suitable for most purposes regarding image acquisition, processing or even recognition of microscopic objects. A direct comparison between the same FOV acquired on both the RaCAM camera (Sony IMX477 sensor), and a 712 Mono Zeiss Camera (Sony IMX304 sensor) shows subtle to no difference, as most of the resolution is determined by the optical quality of the host microscope and lenses (Fig. 6). Of course, the difference in pricing is largely due to other features implemented in the hi-end cameras (e.g., rolling shutter for the RaCAM vs global shutter for most hi-end camera, more advanced and expensive, but not mandatory for regular microscopy).

Time needed to perform the whole workflow for a single FOV image varies depending on the number of object images that are segmented and need to be identified and board version (from 1 to 16 GB of RAM, the higher the better). Individually, the image capture step is almost instantaneous (excluding a 5 sec. delay, whose duration can however be increased or decreased, to perform a full-screen focal adjustment before the FOV image is actually captured). The image processing is about 4 sec. per FOV, and includes saving of all generated individual object images. Finally, time needed to perform image recognition is

https://doi.org/10.5194/egusphere-2026-2075
Preprint. Discussion started: 22 April 2026

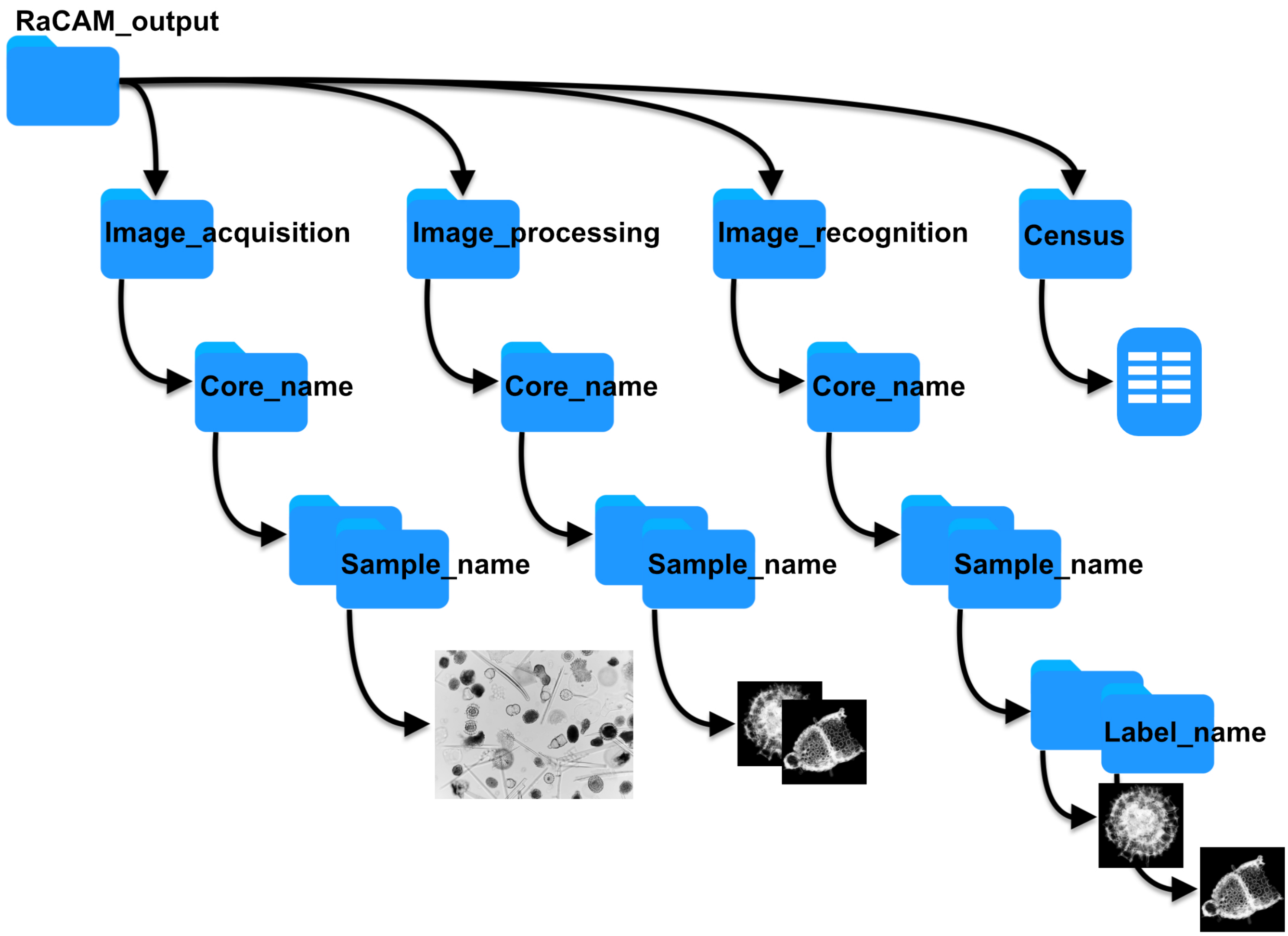

Figure 5. File saving directory structure for the original FOV images, segmented object images, and labeled object images.

about 1-2 sec. per individual object image. For example, overall, the whole workflow would last for about 15 sec (for 5 objects detected), 20-25 sec (10 objects), 45 sec (20 objects), and so on. Generating a census data file is almost instantaneous.

3.5 Field testing the camera

For an accurate and representative field testing of the camera, we decided to use it routinely during the third field season for the SWAIS2C programme (Sensitivity of the West Antarctic Ice Sheet to 2 degrees Celsius of warming), a six weeks field season on ice at Crary Ice Rise, located 700 km off Scott Base that aimed to recover a 200 m+ sediment core from below the ice sheet. The camera was mounted on a Nikon Alphaphot microscope for image capture of micropaleontological smear slide images for real-time microfossils assessment (diatoms, silicoflagellates, ebridian cysts) and biostratigraphic purposes during the drilling operations.

https://doi.org/10.5194/egusphere-2026-2075
Preprint. Discussion started: 22 April 2026

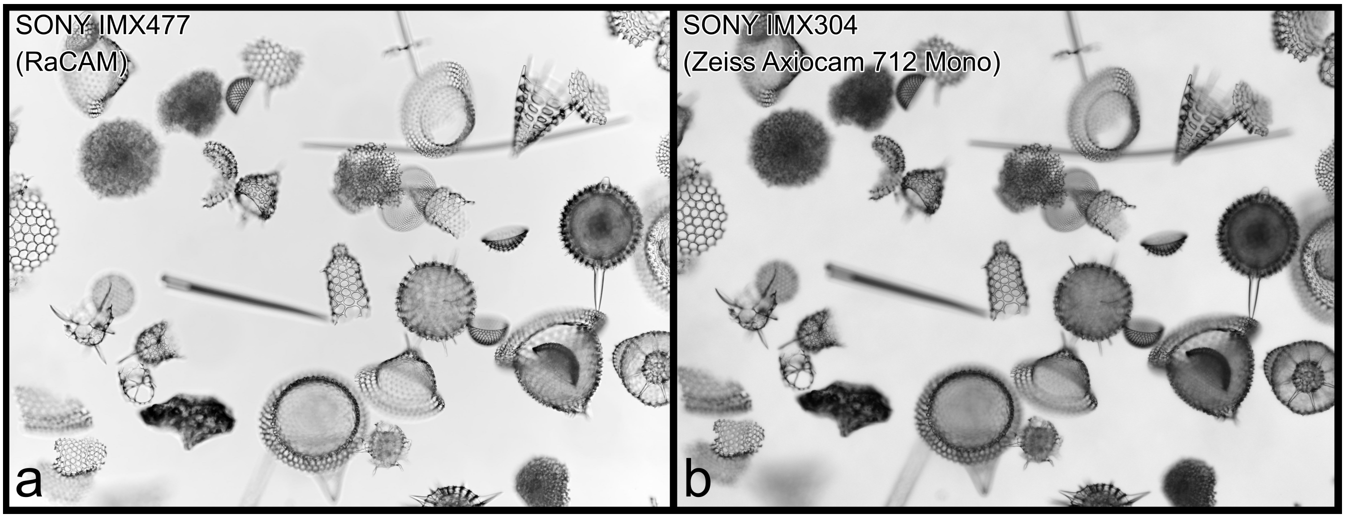

Figure 6. Image quality comparison for the same field of view (Radiolarian and diatoms Paleocene fauna, sample ODP181-1121B-5470cm) captured with the RaCAM camera (**a**), and a Zeiss 712 Mono camera (**b**).

For about two weeks, the camera was used continuously for 12-14 hours a day as part of an upgraded biostratigraphic workflow, for live-monitoring of the drilling process by performing image acquisition of smear slides prepared from samples regularly recovered from every core section. When microfossils were present, about 20 FOV images per sample were acquired, and automatically processed to segment every particle (biogenic or not), which were then automatically labeled and saved using a CNN trained for inference of Quaternary to Pliocene Antarctic marine diatom images.

As the recovered samples were primarily Late to Early Miocene in age, most of the taxa imaged and labeled were not learned yet by our trained CNN. However, most of the diatom specimens identified by the camera were assigned to morphologically and phylogenetically close taxa (e.g., species of the same genus such as *Actinocyclus*, *Fragilariopsis* and *Thalassiosira* species). As the CNN training (performed on another computer), and accuracy are independent from the camera itself, and only dependent on the image database it was trained on, we consider that no technical issue was encountered by the camera. The whole automated “image acquisition, image processing, and image recognition” workflow is thus fully operational. For future field work, new CNNs will be trained for other time intervals, and loaded on the camera in order to cover a longer biostratigraphically significant time bin.

As the camera operates using a built-in micro-computer, allowing for other operations such as using spreadsheet, text document, web browser, and image acquisition, processing and recognition, it prevents the need for a separate computer (Fig. 7). With an average power consumption of about 3 W (about 6W with full CPU load), the RaCAM is very energy efficient and can be operated on field, or anywhere else, using a portable power bank (and together with the USB-powered Raspberry Pi screen).

https://doi.org/10.5194/egusphere-2026-2075
Preprint. Discussion started: 22 April 2026

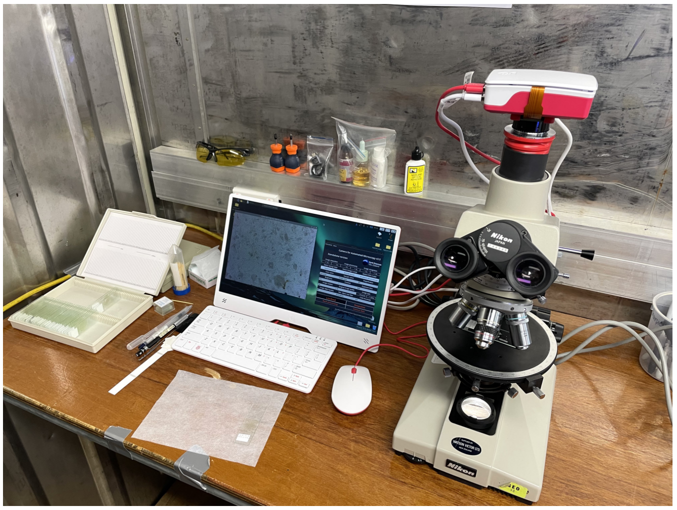

Figure 7. RaCAM camera mounted on a Nikon Alphaphot during the SWAIS2C 2025-2026 season, Crary Ice Rise.

4 Conclusions

We developed an affordable and fully automated camera that can be plugged into the C-mount thread of any microscope such as transmitted light or stereoscopic microscopes. At a cost of about ∼125 to ∼225 EUR (depending on the board version), freely available OS, software and packages, and free guides to build it yourself, the RaCAM camera allows any user to use and improve an affordable automated microscopy workflow, from image acquisition, to image processing, image recognition using trained CNN models, and generation of census data for any microscopic objects investigated using a user-friendly interface.

Installation procedure, RaCAM software, 3D-model files are all available for free as supplementary material and at: https://github.com/microfossil/particle-classification-onnx. Our next steps for the RaCAM project are: (1) to develop object detection in our automated workflow (on top of our current image segmentation and classification workflow) using YOLO; and (2) to develop a motorised stage that would fit most microscopes and be directly controlled by the integrated computer board and coupled with the RaCAM automated microscopy workflow.

Code availability. RaCAM software and files, installation procedure, STL files to print camera sensor adaptors and cases are available at: https://github.com/microfossil/particle-classification-onnx and as supplementary materials.

RaCAM software is a free software (developed using the Python programming language), used to perform image acquisition (using rpicam-apps developed by © Raspberry Pi Ltd), image processing and segmentation (using the ImageJ software) and image recognition using CNNs.

Copyright © 2026 Martin Tetard

This program is free software: you can redistribute it and/or modify it under the terms of the GNU General Public License as published by the Free Software Foundation, either version 3 of the License, or (at your option) any later version.

https://doi.org/10.5194/egusphere-2026-2075
Preprint. Discussion started: 22 April 2026

This program is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU General Public License for more details.

You should have received a copy of the GNU General Public License along with this program. If not, see https://www.gnu.org/licenses/.

Data availability. RaCAM_3D_files.zip © 2026 by Martin Tetard is licensed under CC BY-NC-SA 4.0. To view a copy of this license, visit https://creativecommons.org/licenses/by-nc-sa/4.0/

Author contributions. MT designed the experiment, performed its technical aspects, including adaptor designing, the image acquisition, processing and census data parts of the RaCAM software, and write the first draft of the manuscript. RM developed CNN training and developed the recognition part of the RaCAM software.

Competing interests. The authors declare that no competing interests are present.

Acknowledgements. This study and camera parts were funded by the Antarctic Science Platform (ASP) and SWAIS2C Programme. This work was supported by the Natural Environment Research Council, Alfred-Wegener-Institute Helmholtz Centre for Polar and Marine Research, Federal Institute for Geosciences and Natural Resources, National Science Foundation (NSF-2035029, 2034719, 2034883, 2034990, 2034999, 2035035, and 2035138), German Research Foundation (grants KU 4292/1-1, MU 3670/3-1, KL 3314/4-1), Istituto Nazionale di Geofisica e Vulcanologia, Korea Polar Research Institute, National Institute of Polar Research, Antarctic Science Platform (ANTA1801), LIAG Institute for Applied Geophysics, AuScope, and the Australian and New Zealand IODP Consortium. Logistical support comes from Antarctica New Zealand (K862A) in collaboration with the United States Antarctic Program. Drilling is funded and supported by the International Continental Scientific Drilling Program. Te Herenga Waka—Victoria University of Wellington manages drilling and core recovery. ESNZ Science provides project management. MT contribution was also supported by the New Zealand Ministry of Business, Innovation and Employment through the Kaitiakitanga ki Te Riu-a-Māui (KTRM) and Changing Climate and Environment (CCE) SSIF Programmes (Contract C05X1702). Many thanks to the editor, and reviewers for reviewing the manuscript.

https://doi.org/10.5194/egusphere-2026-2075
Preprint. Discussion started: 22 April 2026

References

Beaufort, L., and Dollfus, D.: Automatic recognition of coccoliths by dynamical neural networks, Mar. Micropaleontol., 51, 57–73, https://doi.org/10.1016/j.marmicro.2003.09.003, 2004.

Bourel, B., Marchant, R., de Garidel-Thoron, T., Tetard, M., Barboni, D., Gally, Y., and Beaufort, L.: Automated recognition by multiple convolutional neural networks of modern, fossil, intact and damaged pollen grains, Comput. and Geosci., 140, 104498, https://doi.org/10.1016/j.cageo.2020.104498, 2020.

Carlsson, V., Danelian, T., Boulet, P., Devienne, P., Laforge, A., and Renaudie, J.: Artificial intelligence applied to the classification of eight middle Eocene species of the genus *Podocyrtis* (polycystine radiolaria), J. Micropalaeontol., 41, 165–182, https://doi.org/10.5194/jm-41-165-2022, 2022.

Carlsson, V., Danelian, T., Tetard, M., Meunier, M., Boulet, P., Devienne, P., and Ventalon, S.: Convolutional neural network application on a new middle Eocene radiolarian dataset, Mar. Micropaleontol., 183, 102268, https://doi.org/10.1016/j.marmicro.2023.102268, 2023.

Daood, A., Ribeiro, E., and Bush, M.: Sequential recognition of pollen grain Z-stacks by combining CNN and RNN, in: The Thirty-First International Florida Artificial Intelligence Research Society Conference, Melbourne, Florida, USA, pp. –13, 2018.

Dollfus, D., and Beaufort, L.: Fat neural network for recognition of position-normalised objects, Neural networks, 12, 553–560, https://doi.org/10.1016/S0893-6080(99)00011-8, 1999.

Gimenez, B., Joannin, S., Pasquet, J., Beaufort, L., Gally, Y., de Garidel-Thoron, T., Combourieu-Nebout, N., Bouby, L., Canal, S., Ivorra, S., Limier, B., Terral, J.-F., Devaux, C., and Peyron, O.: A user-friendly method to get automated pollen analysis from environmental samples, New Phytol, 243, 797–810, https://doi.org/10.1111/nph.19857, 2024.

Govindankutty Menon, A., Davis, C.V., Nurnberg, D. Nomaki, H., Salonen, I., Schmiedl, G., and Glock, N.: A deep-learning automated image recognition method for measuring pore patterns in closely related bolivinids and calibration for quantitative nitrate paleo-reconstructions, Sci. Rep., 13, 19628, https://doi.org/10.1038/s41598-023-46605-y, 2023.

Grant, S. D., Richford, K., Burdett, H. L., McKee, D., and Patton, B. R.: Low-cost, open-access quantitative phase imaging of algal cells using the transport of intensity equation, R. Soc. Open Sci., 7, 191921, https://doi.org/10.1098/rsos.191921, 2020.

Hoffecker, E.: Local Innovation: what it is and why it matters for developing economies, D-Lab Working Papers: NDIR Working Paper 01, Cambridge: MIT D-Lab, 2018.

Hsiang, A. Y., Brombacher, A., Rillo, M. C., Mleneck-Vautravers, M. J., Conn, S., Lordsmith, S., Jentzen, A., Henehan, M. J., Metcalfe, B., Fenton, I. S., Wade, B. S., Fox, L., Meilland, J., Davis, C. V., Baranowski, U., Groeneveld, J., Edgar, K. M., Movellan, A., Aze, T., Dowsett, H. J., Giles Miller, C., Rios, N., and Hull, P. M.: Endless Forams: >34,000 modern planktonic foraminiferal images for taxonomic training and automated species recognition using convolutional neural networks, Paleoceanogr., 34, 1157–1177, https://doi.org/10.1029/2019PA003612, 2019.

Itaki, T., Taira, Y., Kuwamori, N., Saito, H., Ikehara, M., and Hoshino, T.: Innovative microfossil (radiolarian) analysis using a system for automated image collection and AI-based classification of species, Sci. Rep., 10, 21136, https://doi.org/10.1038/s41598-020-77812-6, 2020.

Johnston, S. J., and Cox, S. J.: The Raspberry Pi: A Technology Disrupter, and the Enabler of Dreams. Electronics, 6, 51, https://doi.org/10.3390/electronics6030051, 2017.

Lynn, H.: A low-cost, open-source, computer-assisted microscope, [online] Available from: https://www.raspberrypi.com/news/a-low-cost-open-source-computer-assisted-microscope/ (Accessed 20 December 2025), 2019.

https://doi.org/10.5194/egusphere-2026-2075
Preprint. Discussion started: 22 April 2026

Mimura, K., Nakamura, K., Yasukawa, K., Sibert, E. C., Ohta, J., Kitazawa, T., and Kato, Y.: Applicability of object detection to microfossil research: Implications from deep learning models to detect microfossil fish teeth and denticles using YOLO-v7, ESS, 11, e2023EA003122, https://doi.org/10.1029/2023EA003122, 2024.

Mimura, K., Itaki, T., Kataoka, H., and Miyakawa, A.: Classifying microfossil radiolarians on fractal pre-trained vision transformers, Sci. Rep., 15, 7189, https://doi.org/10.1038/s41598-025-90988-z, 2025.

Marchant, R., Tetard, M., Pratiwi, A., and de Garidel-Thoron, T.: Automated analysis of foraminifera fossil records by image classification using a convolutional neural network, J. Micropalaeontol., 39, 183–202, https://doi.org/10.5194/jm-39-183-2020, 2020.

Martinsen, I., Ricaud, B., Wade, D., Kolbjørnsen, and O., Godtliebsen, F.: The Fossil Frontier: An answer to the 3-billion fossil question, Artif. Intell. Geosci., 7, 100184, https://doi.org/10.1016/j.aiig.2025.100184, 2026.

Mitra, R., Marchitto, T. M., Ge, Q., Zhong, B., Kanakiya, B., Cook, M. S., Fehrenbacher, J. S., Ortiz, J. D., Tripati, A., and Lobaton, E.: Automated species-level identification of planktic foraminifera using convolutional neural networks, with comparison to human performance, Mar. Micropaleontol., 148, 1–14, https://doi.org/10.1016/j.marmicro.2019.01.005, 2019.

Nowosad, M.: Build your own 10.000€microscope for around 150€, [online] Available from: https://martinnowosad.medium.com/turning-my-trinocular-microscope-into-a-smart-digital-lab-camera-for-around-150-2dfe13afcb1c (Accessed 20 December 2025), 2025.

Piva, A., Raimondi, L., Rasca, E., Kozmanyan, A., and De Matteis, M.: A machine learning application for the automatic recognition of planktonic foraminifera in thin sections, Mar. Pet. Geol., 166, 106911, https://doi.org/10.1016/j.marpetgeo.2024.106911, 2024.

Richmond, T., Cole, J., Dangler, G., Daniele, M., and Marchitto, T., Lobaton, E.: Forabot: Automated planktic foraminifera isolation and imaging, G3, 23, e2022GC010689, https://doi.org/10.1029/2022GC010689, 2022.

Schneider, C. A., Rasband, W. S., and Eliceiri, K. W.: NIH Image to ImageJ: 25 years of image analysis, Nat. Methods, 9, 671–675, https://doi.org/10.1038/nmeth.2089, 2012.

Schneidereit, D., Kraus, L., Meier, J. C., Friedrich, O., and Gilbert, D. F.: Step-by-step guide to building an inexpensive 3D printed motorized positioning stage for automated high-content screening microscopy, Biosens. Bioelectron., 92, 472–481, https://doi.org/10.1016/j.bios.2016.10.078, 2017.

Tetard, M., Marchant, R., Cortese, G., Gally, Y., Thibault de Garidel-Thoron, and Beaufort, L.: Technical note: A new automated radiolarian image acquisition, stacking, processing, segmentation and identification workflow, Clim. Past, 16, 2415–2429, https://doi.org/10.5194/cp-16-2415-2020, 2020.

Tetard, M., Carlsson, V., Meunier, M., and Danelian, T.: Merging databases for CNN image recognition, increasing bias or improving results? Mar. Micropaleontol. 185, 102296, https://doi.org/10.1016/j.marmicro.2023.102296, 2023.

Webb, H.: Raspberry Pi Microscopy, [online] Available from: http://www.microscopy-uk.org.uk/mag/artfeb17/hw-RaspberryPi.html (Accessed 20 December 2025), 2017.

William, W.: Openflexure Microscope V7.0.0-beta1 released, [online] Available from: https://openflexure.discourse.group/t/openflexure-microscope-v7-0-0-beta1-released/1098 (Accessed 20 December 2025), 2022.

William, W.: Minimal manual version of the Openflexure microscope, [online] Available from: https://openflexure.discourse.group/t/minimal-manual-version-of-the-openflexure-microscope/1257 (Accessed 20 December 2025), 2023.

Whittaker, A.: Raspberry Pi High Quality Camera powers up homemade microscope by DIY-Maxwell, [online] Available from: https://www.raspberrypi.com/news/raspberry-pi-high-quality-camera-powers-up-homemade-microscope/ and https://github.com/IBM/MicroscoPy (Accessed 20 December 2025), 2020.

https://doi.org/10.5194/egusphere-2026-2075
Preprint. Discussion started: 22 April 2026

Wong, A. P. S., and Coauthors: Argo data 1999–2019: Two million temperature-salinity profiles and subsurface velocity observations from a global array of profiling floats, Front. Mar. Sci., 7, 700, https://doi.org/10.3389/fmars.2020.00700, 2020.

Zhang J, Vieira, D. N., Cheng, Q., Zhu, Y., Deng, K., Zhang, J., Qin, Z., Sun, Q., Zhang, T., Ma, K., Zhang, X., and Huang, P.: DiatomNet v1.0: A novel approach for automatic diatom testing for drowning diagnosis in forensically biomedical application, Comput. Methods Programs Biomed., 232, 107434, https://doi.org/10.1016/j.cmpb.2023.107434, 2023.