

Automated segmentation of synchrotron-scanned fossils

Melanie A. D. During^{1*}, Jordan K. Matelsky^{2,3*}, Fredrik K. Gustafsson⁴, Dennis F. A. E. Voeten^{1,5}, Donglei Chen¹, Brock A. Wester³, Konrad P. Kording^{2,6}, Per E. Ahlberg¹, Thomas B. Schön⁷

1 Department of Organismal Biology, Uppsala University, Uppsala, Sweden

2 Department of Bioengineering, University of Pennsylvania, Philadelphia PA, USA

3 Research and Exploratory Development, Johns Hopkins University Applied Physics Laboratory, Laurel MD, USA

4 Department of Medical Epidemiology and Biostatistics, Karolinska Institutet, Stockholm, Sweden

5 Frisian Museum of Natural History, Leeuwarden, Netherlands

6 Department of Neuroscience, University of Pennsylvania, Philadelphia PA, USA

7 Department of Information Technology, Uppsala University, Uppsala, Sweden

<https://zoobank.org/8CC1E725-4EA5-489F-AF3A-CA4BA2E30214>

Corresponding authors: Melanie A. D. During (melanie.during@ebc.uu.se); Jordan K. Matelsky (matelsky@seas.upenn.edu)

Academic editor: Alexander Schmidt ♦ Received 15 October 2024 ♦ Accepted 6 January 2025 ♦ Published 6 March 2025

Abstract

Computed tomography has revolutionised the study of the internal three-dimensional structure of fossils. Historically, fossils typically spent years in preparation to be freed from the enclosing rock. Now, X-ray and synchrotron tomography reveal structures that are otherwise invisible, and data acquisition can be fast. However, manual segmentation of these 3D volumes can still take months to years. This is especially challenging for resource-poor teams, as scanning may be free, but the computing power and (AI-assisted) segmentation software required to handle the resulting large data sets are complex to use and expensive.

Here we present a free, browser-based segmentation tool that reduces computational overhead by splitting volumes into small chunks, allowing processing on low-memory, inexpensive hardware. Our tool also speeds up collaborative ground-truth generation and 3D visualisation, all in-browser. We developed and evaluated our pipeline on various open-data scans of differing contrast, resolution, textural complexity, and size. Our tool successfully isolated the *Thrinaxodon* and *Broomistega* pair from an Early Triassic burrow. It isolated cranial bones from the Cretaceous acipenseriform *Parapsephurus willybemisi* on both 45.53 μm and 13.67 μm resolution (voxel size) scanning data. We also isolated bones of the Middle Triassic sauropterygian *Nothosaurus* and a challenging scan of a squamate embryo inside an egg dating back to the Early Cretaceous. Our tool reliably reproduces expert-supervised segmentation at a fraction of the time and cost, offering greater accessibility than existing tools. Beyond the online tool, all our code is open source, enabling contributions from the palaeontology community to further this emerging machine-learning ecosystem.

Key Words

AI-segmentation, Machine Learning, Open Source, Open Access, Propagation Phase-Contrast Synchrotron Radiation Micro-Computed Tomography (PPC-SR μ CT), Random Forest

* These authors contributed equally to this work.

Introduction

Fossilisation is rare. Biological remains require very specific circumstances to be preserved over a long period of time. Subsequently, these fossils need to be found and appropriately extracted for academic study. Physical preparation of fossils used to be the only way to explore fossil contents. Yet preparation is not without risks, as it can damage the bone surface, and it will remove potential soft tissues that are not always recognisable upon exposure. Furthermore, it is not guaranteed to reveal all of the anatomy of interest. Museum curators are understandably hesitant to allow for destructive analyses or extensive preparation of delicate structures. Recent technological advances, such as Computed Tomography (CT) in general and Propagation Phase-Contrast Synchrotron Radiation Micro-Computed Tomography (PPC-SR μ CT; Tafforeau et al. 2006) in particular, have made it possible to study fossils at sub-micron resolution, revealing parts of the anatomy that were previously hidden by matrix, and enabling the study of internal anatomical features or preserved soft tissue without damaging the fossils. This innovation has dramatically increased our ability to interrogate otherwise impenetrable palaeontological mysteries.

Despite these advantages, the 3D volumes produced by X-ray tomography (XRT) require substantial digital storage and computing power. The subsequent segmentation and modelling workflows thus tend to demand substantial funds that are not universally available to research groups and can take so much manual labour that projects remain unfinished. Commercial endeavours offer machine learning-based segmentation as a solution to these challenges and have already made a significant impact in the field (Edie et al. 2023), but these tools are typically designed for industrial applications and are therefore fine-tuned for neither synchrotron nor palaeontological data. To meet the need for palaeontology-focused, accessible, and democratised XRT analysis software, our team developed an online segmentation pipeline for fossils, trained on publicly available synchrotron data, and offered it as a free service to the public (Fig. 1).

Here, we present the *ml4paleo* software suite, a Python package that combines traditional machine learning tools, neural image segmentation tools, and batch processing tools to segment large-scale palaeontological XRT data volumes. We share a simple “online-learning” web interface for machine-guided, 2D slice-based image annotation to aid in training small models without downloading new software. Our solution can be operated with minimal technical expertise and runs on commodity computing hardware. Our codebase is an open source and we encourage community feedback and contributions.

Automated segmentation is a major ongoing challenge across the XRT community broadly (Tanti et al. 2021 and Hider et al. 2022). The specific challenges in palaeontology, such as the variation in contrast through an image stack, and high absorption caused by metallic inclusions, reflect similar challenges that other fields face with their

own XRT workloads. We therefore intend for our tool to serve as a model and inspiration for a wide range of fields, such as neuroscience, histology, and earth sciences.

Materials and methods

Materials

We leveraged five public datasets that are available online in the paleo.esrf.eu database, a public resource for palaeontological volumes scanned at the European Synchrotron Radiation Facility (ESRF).

The datasets used in this study represent a range of fossil types and scanning conditions, each presenting its unique challenges to segmentation. The Burrow dataset, from the Early Triassic Karoo Basin in South Africa, contains complete skeletons of a *Thrinaxodon* and a *Broomistega* and was scanned at 45.5 μm resolution by Fernandez et al., (2013). This dataset benefits from a relatively clear contrast between the fossils and the surrounding matrix, allowing for high segmentation accuracy. In contrast, the Middle Triassic *Nothosaurus* skull from the Muschelkalk of the Netherlands scanned at 12.82 μm resolution (Voeten et al. 2018) presents lower contrast. This makes it more difficult to distinguish between fossilised bone and matrix. The *Nothosaurus* stack was also post-processed into a recoded version of the three-dimensional data based on local texture complexity to reveal low-contrast features (Voeten et al. 2018). The end-Cretaceous paddlefish *Parapsephurus willybemisi*, from the Tanis deposit in North Dakota, was scanned at 45.53 μm and 13.67 μm resolution (During et al. 2022 and Hilton et al. 2023). These datasets highlight the role of scan resolution in segmentation accuracy. The higher-resolution $\sim 13 \mu\text{m}$ dataset allows for the segmentation of more detailed phenomena, while the lower-resolution $\sim 45 \mu\text{m}$ dataset does not, as key anatomical features become more difficult to discern. Finally, an Early-Cretaceous squamate embryo inside an egg from Thailand (Phu Phok) was scanned at 5.06 μm resolution by Fernandez et al. (2015). This dataset presents a unique challenge due to the fine structures, compounded by low contrast, leading to a significant challenge in segmentation for both humans and machine learning-based methods. For additional details about the subsamples of these datasets used for our evaluation, see Table 1.

Web application architecture

The *ml4paleo* web application comprises four components; one main application server, and three task runners that operate on an on-disk queue.

Application server: The web application server is written in Flask (Grinberg 2018). The server can use an on-disk serialisation format for storing dataset metadata, or it can be configured to run a standalone database

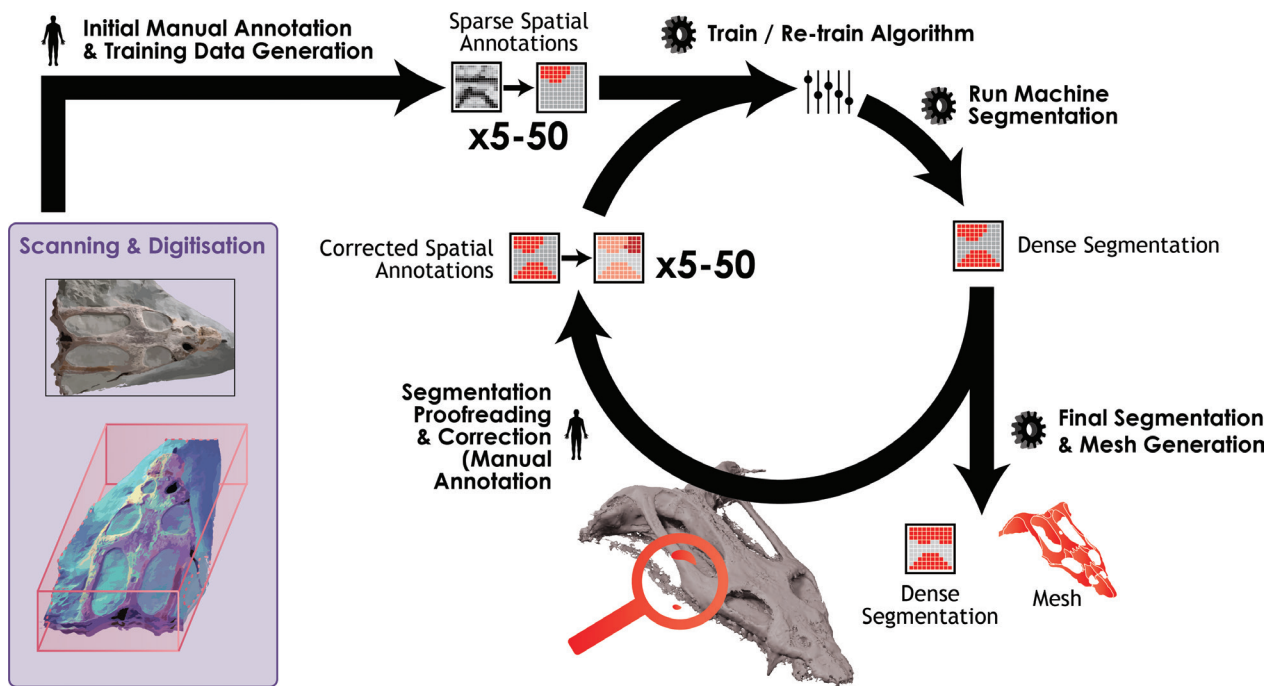


Figure 1. An overview of the pipeline proposed here. Raw X-ray image volumes (left) are iteratively segmented through collaborative human-machine teaming. Machine-guided annotation in our web application yields training data for the segmentation models. The segmentation model is applied to the whole dataset through batch processing, taking advantage of chunked data storage paradigms. This process can be repeated until the model meets human-defined quality benchmarks. Finally, a high-resolution segmentation mask can be exported, alongside image renders and meshes suitable for rendering in 3D software.

Table 1. Datasets used in this study.

Dataset nickname	Taxa	Age	Location	Dataset Size (pixels)	Resolution	Reference
Paddlefish ~45um	<i>Parapsephurus willybemisi</i>	Late Cretaceous	Tanis Deposit (ND, USA)	1771,1117,200	45.53 μm	During et al. (2022)
Paddlefish ~13um	<i>Parapsephurus willybemisi</i>	Late Cretaceous	Tanis Deposit (ND, USA)	3815,3815,200	13.67 μm	During et al. (2022)
Burrow	<i>Thrinaxodon, Broomistega</i>	Early Triassic	Karoo Basin (South Africa)	1685,3043,200	45.5 μm	Fernandez et al. (2013)
Phu Phok	Anguimorph embryo	Early Cretaceous	Phu Phok (Thailand)	1939,2206,200	5.06 μm	Fernandez et al. (2015)
Nothosaur	<i>Nothosaurus marchicus</i>	Middle Triassic	Winterswijk (The Netherlands)	1410,1410,200	12.82 μm	Voeten et al. (2018)

daemon. By default, our application uses a standalone JSON file as a metadata store to reduce the software dependencies of running a minimal deploy.

Conversion queue: Researchers commonly receive concatenated image stacks as output from the synchrotron facility. Such image stacks tend to be too large and inefficient to access for common machine learning workflows. Thus, the first memory-intensive stage of processing is to convert this inefficient data format into a chunked data format. The user can easily upload their dataset in most common formats, including image stacks and some common chunked formats. Chunking high-dimensional data involves re-organising a large dataset of 2D images into smaller, more compact 3D (volumetric) or 4D (volumetric time series) pieces, known as chunks. These chunks can then be treated as independent small volumes or manipulated in spatial sequence. This conversion

enables the *ml4paleo* tools to operate on sub-volume cut-outs without loading complete slices into memory. It is this memory efficiency that enables our infrastructure to work equally well on low-performance and high-grade computers alike.

Model architectures

As a core design principle for this software package, we aimed to select default algorithms that impose minimal restrictions on user hardware, ensuring accessibility for a wide range of systems. In other words, our goal was to accommodate the lowest common denominator in terms of hardware capabilities. This required our segmentation algorithms to be compatible with consumer-grade hardware while operating at the necessary spatial scales.

Two primary decisions emerged from this requirement: first, the dimensionality of the segmentation process, and second, the selection of machine learning models, based on both computational demands and dimensionality considerations.

While 2D stacked segmentation, where individual slices are segmented and later reconstituted into 3D, can deliver acceptable results in many cases, it is inferior to true 3D segmentation. The latter benefits from the inclusion of additional spatial context within the 3D data stack. However, memory limitations significantly reduce the efficiency of 3D segmentation, as its computational complexity scales with N^3 compared to N^2 for 2D segmentation. Given that X-ray tomography (XRT) data is isotropic, with uniform voxel sizes across all dimensions, 2D models have proven sufficient for our purposes thus far. Nevertheless, our codebase includes 3D models, allowing users to switch between 2D and 3D approaches as needed (see Data & Code Availability).

As the default segmentation algorithm, we utilise a simple Random Forest (RF) model trained on heuristic-based pixel features (Pedregosa et al. 2011). This is implemented using the "skimage.feature.multiscale_basic_features" module with default parameters, all of which are adjustable through the administrator settings of the web application. The feature extractor relies on basic image properties such as texture, intensity, and edge detection, progressively applying Gaussian filters to estimate these features on "downsampled" versions of the images.

For training efficiency, we optionally provide a parameter to subsample pixels, using every n th pixel for training rather than all annotated pixels. By default, 50% of the available image/segmentation pairs are used during the training process.

Our codebase also includes a convolutional residual U-Net (Ronneberger et al. 2015), which requires a GPU for efficient training, though it otherwise operates with modest computational requirements. These tools are implemented through the "ml4paleo.segmentation.Segmenter3D" interface. Both this and the "ml4paleo.segmentation.Segmenter2D" interface are designed to support easy integration of more advanced or specialised segmentation algorithms by third-party contributors.

Data processing pipeline

In order to process arbitrarily large volumes of 3D imagery, it is necessary to decrease the size of the volume that is fed into the pipeline. In volumetric data processing, this is referred to as "chunking" the volume into subvolumes with contiguous areas nearby in memory. Several data standards have been built in a variety of scientific domains to meet this need (Miles et al. 2020; Matelsky et al. 2021; Hider et al. 2022). For most purposes, interchangeable data standards make the distinctions between these chunked formats insignificant, and for this project, we have used the standard "zarr" format (Miles et al. 2020). This chunking technique makes it possible to independently segment

subvolumes of the dataset, which can then be combined in a segmentation fusion step later. In their corresponding *Python* Application Programming Interfaces (APIs), these "datastores" tend to meet the standard numpy-like slicing interface in order to be maximally interoperable with the machine-learning ecosystem (Harris et al. 2020). This also means that data can be lazily loaded from disk so that the majority of the volume can remain compressed and at rest while subvolumes are accessed in memory.

The fusion step can be quite nuanced: in some cases, a single pixel may be justifiably assigned to more than one segmentation type, and therefore it is generally advised to store segmentation as a multi-channel output. While this enables higher-quality segmentation, it comes at the cost of greatly increased storage space, as each segmentation mask must be stored independently. Because our intention was to minimise the computational and storage costs for new projects, we opted here to store single-channel segmentation by default, though this is configurable by the administrator.

Training of segmentation model

We provide a simple web application to generate human expert ground truth to train the segmentation pipeline. This tool (Fig. 2) provides a resizable brush for short annotation tasks of small 2D slices. The user can adjust the contrast and scroll several slices up and down in the stack to provide spatial context during annotation. The tasks are designed to be very short in duration to avoid attentional errors and to make exhaustive annotation feasible on resource-constrained hardware, following design principles from Bishop et al. (2021).

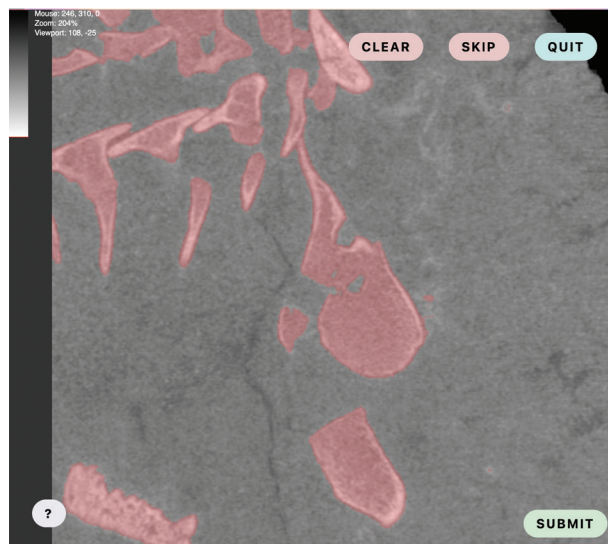


Figure 2. The annotation tool on an excerpt slice of the Burrow dataset (screenshot). The user is given a resizable brush to manually segment fossils from the matrix in a small 2D area. Here, shown in red, the *ml4paleo* automated segmentation has already been run on this slice, so the user only needs to proofread the machine's work, by adding with a brush or removing with an eraser, rather than annotate *de novo*.

Pairs of human-annotated segmentation masks and the original corresponding imagery are then fed to the selected machine learning model for training. By default, *ml4paleo* uses a random forest with local pixel-neighbourhood features (Pedregosa et al. 2011), though the segmentation algorithm in use can be replaced by a simple U-net neural network (Ronneberger et al. 2015) or other machine learning models that implement our “ml4paleo.segmentation.Segmenter3D” interface. A checkpoint of the trained model is then saved for reuse under a unique identifier that contains both the timestamp of training and the specific training samples used. This means that as more data is generated, a library of models can be accumulated to enable analysis and understanding of the relationship between training data volume and segmentation quality.

Segmentation queue: Once a machine learning model has been trained for image segmentation, the dataset is queued for segmentation. In this stage, each worker loads a subvolume of the imagery, performs a dense segmentation, and saves a new segmentation mask volume to disk.

This new layer is identified by the unique model identifier that was generated during the model training; thus, segmentation layers can be uniquely associated with the model that generated them, for provenance and reproducibility.

Mesh queue: As a postprocessing stage after segmentation, a user may opt to download their dataset as a mesh in OBJ or STL formats for rendering in 3D software. This mesh process also operates on one chunk at a time, after which the sub-meshes are stitched into one large mesh.

Visualisation in three dimensions: To visualise the scan as well as the annotations and segmentations, we utilise “Neuroglancer” (Fig. 3), a high-performance, WebGL-based visualisation framework developed by the Google Connectomics Team, designed for efficiently viewing and analysing large-scale volumetric neuroscience datasets (Charles et al. (2020)). A deployment of Neuroglancer can be packaged with deployments of the *ml4paleo* software suite, or any public Neuroglancer application can be used and pointed at *ml4paleo* data volumes through our volumetric API.

Online learning

Once the user has produced at least one saved model — requiring a minimum of one annotated slice — they can optionally allow the machine to produce an initial hypothesis segmentation before each subsequent annotation task. This has the potential to greatly reduce the labour cost of individual annotation tasks, since the machine may produce a nearly complete segmentation mask, depending on the image properties of the underlying volume. Furthermore, the available eraser tool enables users to refine segmented slices, significantly reducing the effort compared to re-segmenting from scratch. Through this machine guidance, the user will become aware of the qualitative performance characteristics of the segmentation model and can monitor its improvement throughout the annotation process.

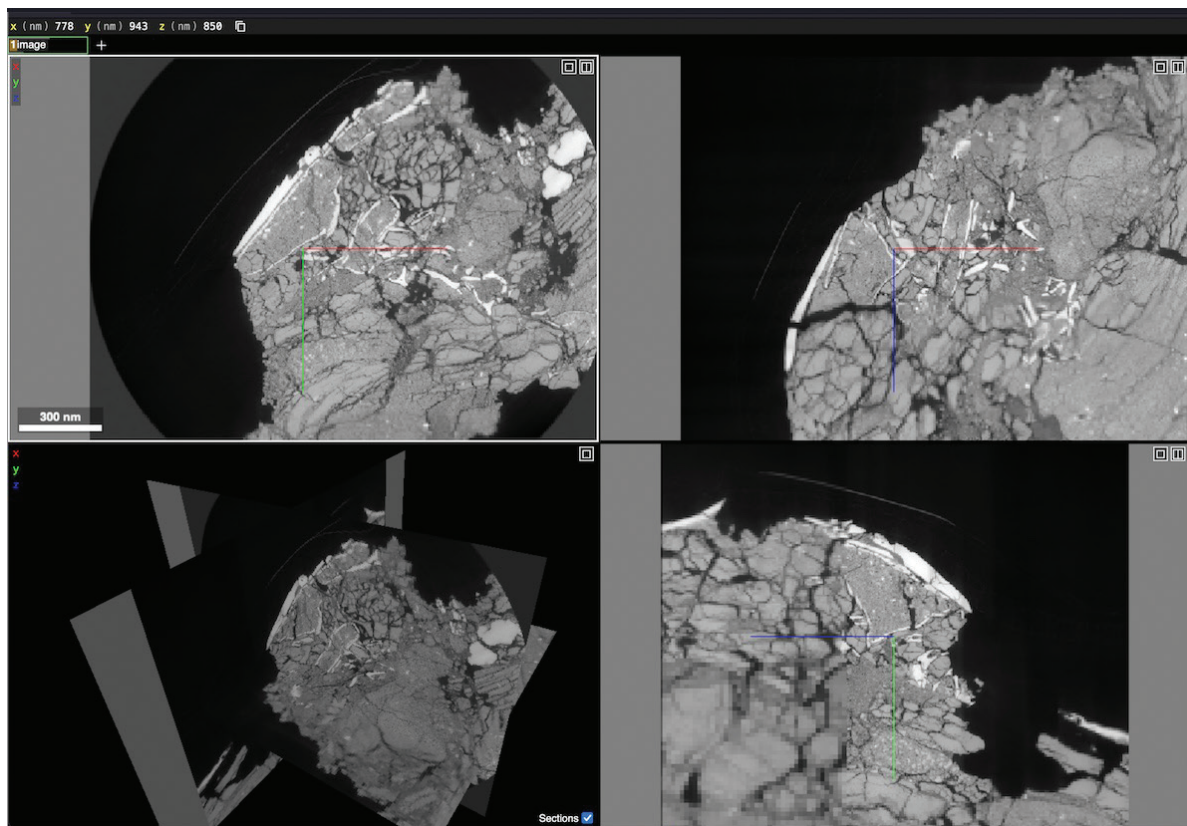


Figure 3. Browser-based visualisation. This stage in the workflow uses *Neuroglancer* to render imagery and annotations. This figure shows a user visualising the *Parapsephurus willybemisi* 13.67 μm resolution scan (During et al. 2022) in interactive 3D in a modern web browser. This tool is included with the *ml4paleo* web suite.

In our approach, for the datasets tested in this study, a total of 20 slices (from patches of full slides) were initially annotated by the same expert palaeontology segmentation author (MD) for each dataset using the annotation web application. However, due to the specific challenges posed by the Phu Phok and Nothosaur datasets, additional annotations were required to improve to the required level of performance. These properties and improvements are reported below.

Fossil embryos, such as those in the Phu Phok dataset, are typically poorly ossified (Fernandez et al. 2015), resulting in significant contrast variations between bones, soft tissues, and matrix. This contrast variability makes it very challenging for the human eye to differentiate structures. The same difficulty is encountered by automated segmentation systems, which struggle with this inconsistent presentation of biologically similar structures. Likewise, the Nothosaur dataset contained a crack running through the fossil’s skull, which created areas of phase contrast that obscured the underlying bone texture. Cracks like these are common in fossil scans and pose difficulties for both human and AI segmentation efforts.

Due to these challenges, after the initial 20 slices were annotated for both datasets, an additional 10 slices were segmented and the model was retrained. A final set of 5 slices was then annotated, resulting in a total of 35 annotated slices for the Nothosaur and Phu Phok Egg datasets to achieve satisfactory segmentation results. In our experiments, switching the default segmentation tool from the random forest to a more sophisticated (neural) model resolved many of these failures, at the cost of additional execution time and GPU requirements.

After the user is satisfied with the model performance, they can deploy the segmentation model on the complete data volume. Below, we provide quantitative evaluation of the *ml4paleo* web application default segmentation, showing performance metrics for leave-one-out cross-validated pairs of human-annotated segmentation and imagery. A visualisation of the ground truth versus the prediction is provided in Fig. 4.

Chunked processing

Because *ml4paleo* datasets are stored in a chunked volumetric data format, it is easy to parallelise an operation across the dataset in small sub-volume increments. Using dataset slicing code borrowed from the electron microscopy neuroscience community (Matelsky et al. 2021), we run the latest model checkpoint from the training stage on each sub-volume of data. When processed in memory, the sub-volumes do not have to match the default size of the chunked format subvolumes on disk ($128 \times 128 \times 128$). Our informal testing suggests that using a larger segmentation sub-volume size of $256 \times 256 \times 256$ can significantly improve throughput without compromising segmentation quality. However, this depends heavily on the available computing resources: memory-constrained environments may benefit from smaller segmentation chunk sizes, while environments with limited CPU cores may perform better with larger chunk sizes.

It is beneficial to have a small overlap of the processed chunks so that the computed pixel-wise features on the edges of each chunk accurately reflect

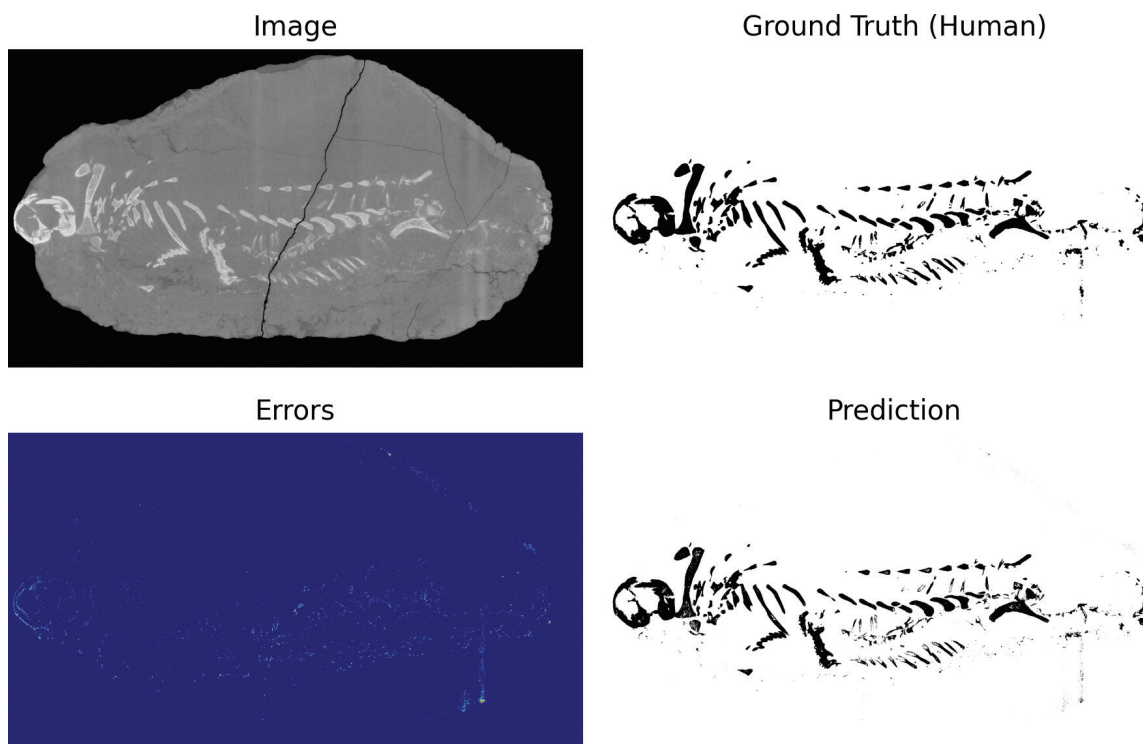


Figure 4. Performance on the burrow dataset. We picture one 2D slice of the original volume (top left), as well as human (top right) and machine annotated segmentation masks (bottom right). On the bottom left, we render the residual errors.

their spatial context (i.e. in the “middle” of the volume, not on an edge). However, we discovered in our tests that this was not always necessary, especially in more easy-to-segment datasets. Samples with a clear brightness contrast between matrix and fossil tended not to need additional context, and fossils distinguished from the matrix only by texture tended to need a broader margin of additional context from neighbouring chunks. A further evaluation and benchmark of this dataset property is left for future work.

Evaluation metrics

To better understand the segmentation performance, we compared it qualitatively to prior segmentations by the original researchers (Fernandez et al. 2013, 2015; Voeten et al. 2018; During et al. 2022) and quantitatively with our own cross-validation evaluation suite. We report the following:

Precision measures the proportion of correctly identified fossil pixels relative to all pixels predicted as fossil by the model. High precision indicates that the model made fewer false positive predictions.

Recall assesses the model’s ability to identify all actual fossil pixels within the dataset. A higher recall means that the model successfully identified more true fossil pixels.

F1-score (Dice coefficient) is the harmonic mean of precision and recall, providing a balanced measure of segmentation quality. It is particularly useful when precision and recall values are unbalanced.

Accuracy measures the proportion of correctly classified pixels (both fossil and background), a high accuracy therefore reflects a low systematic error.

Jaccard index, also called intersection over union, measures similarity between the ground truth and the prediction.

Results

The model’s performance on all five datasets can be found in Table 2 and reported visually in Fig. 5.

Table 2. Performance of our default random forest pixel classifier on a variety of datasets. Values were computed using leave-one-out cross-validation with 2D slice training.

Dataset	Precision	Recall	F1
Burrow	0.861 ± 0.142	0.726 ± 0.154	0.777 ± 0.129
Nothosaurus	0.602 ± 0.227	0.496 ± 0.264	0.522 ± 0.24
Paddlefish ~13um	0.583 ± 0.298	0.492 ± 0.278	0.488 ± 0.257
Paddlefish ~45um	0.718 ± 0.427	0.191 ± 0.187	0.185 ± 0.181
Phu phok	0.401 ± 0.295	0.02 ± 0.014	0.035 ± 0.027

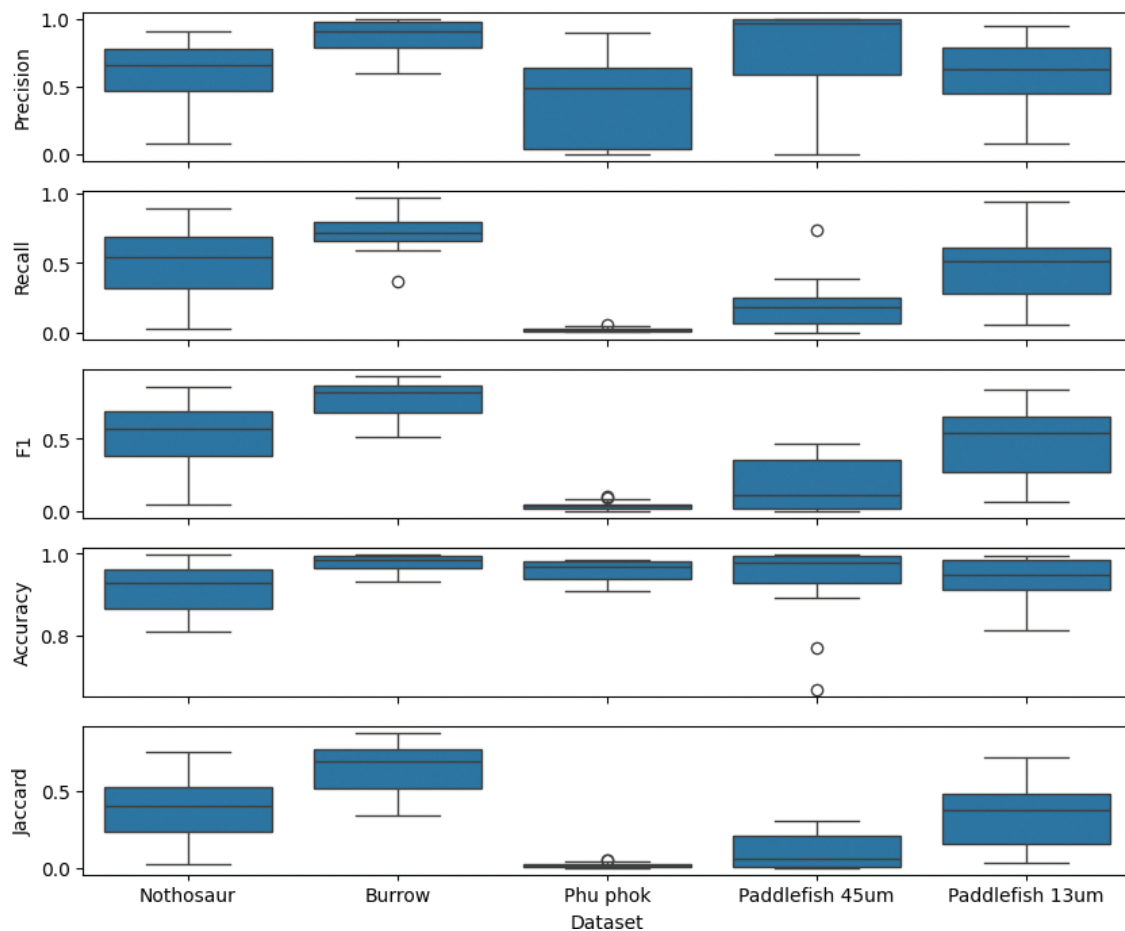


Figure 5. Performance metrics. An illustration of the major differences in segmentation quality between the different datasets, generally due to the inherent difficulty of distinguishing fossil pixels visually (i.e., through contrast or texture). Values are displayed in Table 2.

Burrow data, compared to manual segmentation.

The segmentation model demonstrated strong performance, with high precision and recall, suggesting a robust ability to correctly identify and segment relevant features. The accuracy was also high (0.982), while the Jaccard Index, a measure of similarity between predicted and true masks, was 0.759, indicating a solid overlap between machine predictions and human annotations (Fig. 6).

Paddlefish 13.67 μm and 45.53 μm scans, compared to manual segmentation

Paddlefish $\sim 13 \mu\text{m}$ Dataset (Fig. 7): For this high-resolution scan, performance was moderate, with precision (0.583) and recall (0.492), resulting in an F1-score of 0.488. The segmentation shows a reasonable but not perfect overlap between predicted and true segmentation masks.

Paddlefish $\sim 45 \mu\text{m}$ Dataset (Fig. 8): Performance on this lower-resolution scan was weaker overall, with precision (0.718) and recall (0.191), resulting in an F1-score of 0.185. However, the model achieved high accuracy (0.979), suggesting it effectively avoided false positives, though at the cost of lower sensitivity.

Nothosaur data, compared to manual segmentation

The model's performance declined significantly on this dataset, achieving precision (0.602) and recall (0.496). The low recall suggests that the model struggled to capture true positives. Despite this, accuracy remained reasonably high, likely due to the background dominating the images (Fig. 9).

Phu Phok Egg data, compared to manual segmentation

This dataset posed the greatest challenge, with extremely low precision which was improved by annotating more slices ($n = 20$ 0.232, $n = 35$ 0.401) and recall ($n = 20$ 0.06, $n = 35$ 0.02). Visually, the segmentation quality by the simple random forest was poor overall, reflecting the difficulty encountered by human annotators alike (Fig. 10).

Discussion

The introduction of automated segmentation tools for fossil imaging provides an opportunity to overcome significant bottlenecks in the analysis of large tomographic datasets.

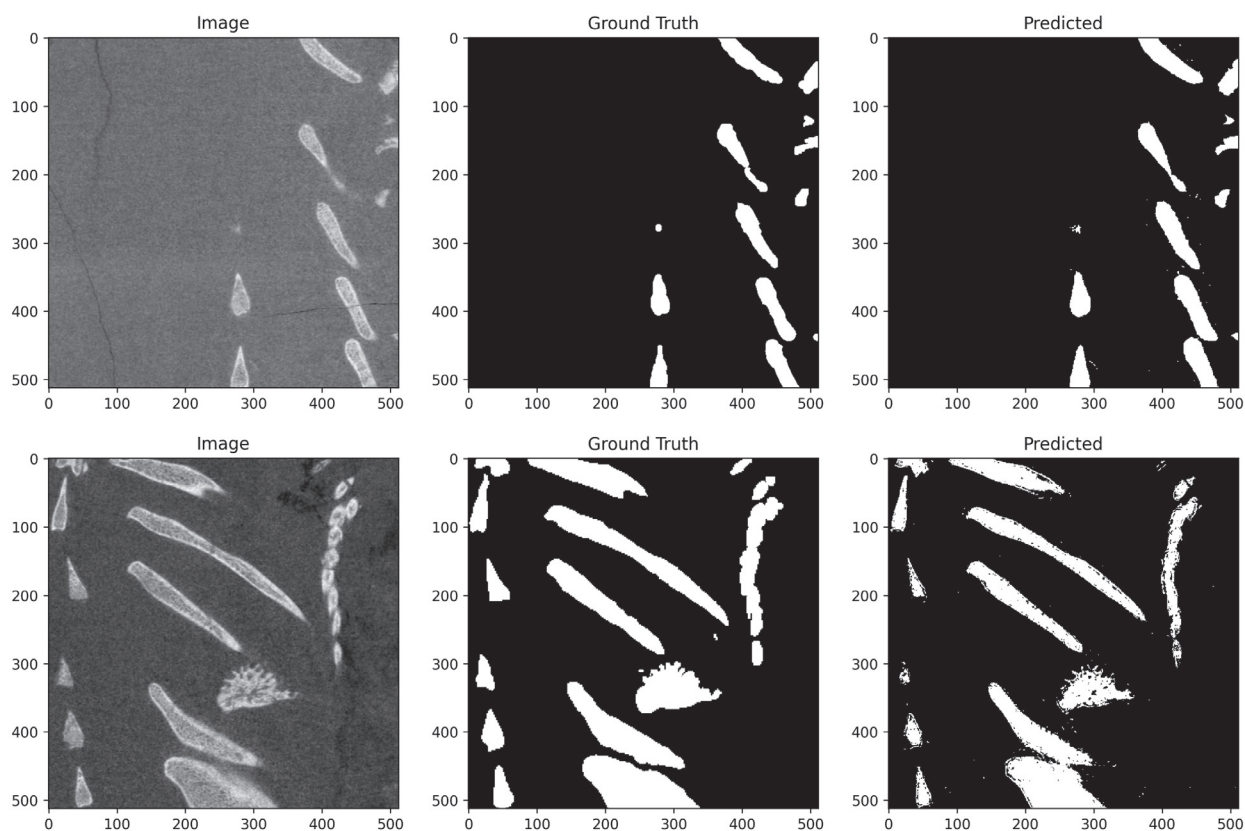


Figure 6. Burrow segmentation samples. The high contrast of the fossil versus the matrix was well-suited for the simple multiscale pixel features used by our default model.

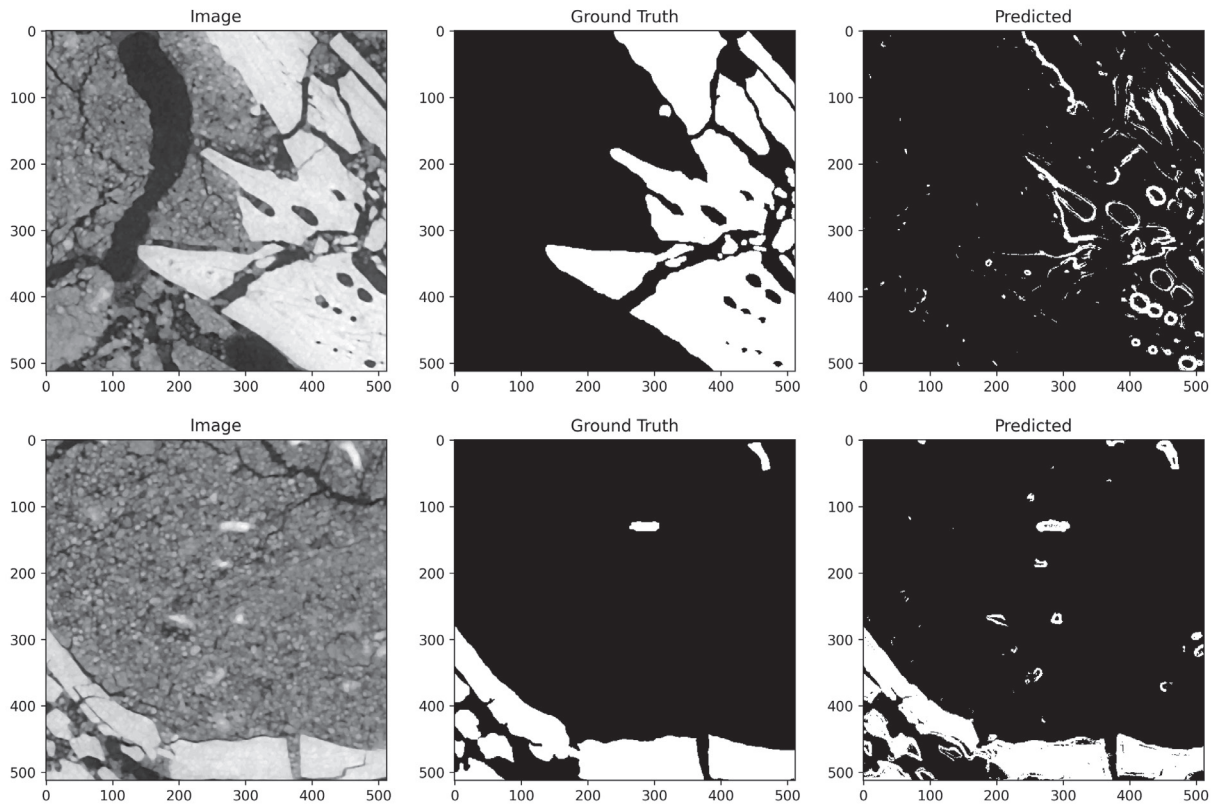


Figure 7. Paddlefish $\sim 13 \mu\text{m}$ segmentation samples. While the segmentation tends to include the right gross morphology, the model notably failed to segment the internal volume of bones correctly. This interestingly has the effect of producing visually “correct” meshes, since the volume of the segmentation is irrelevant to the surface of the generated OBJ or STL meshes.

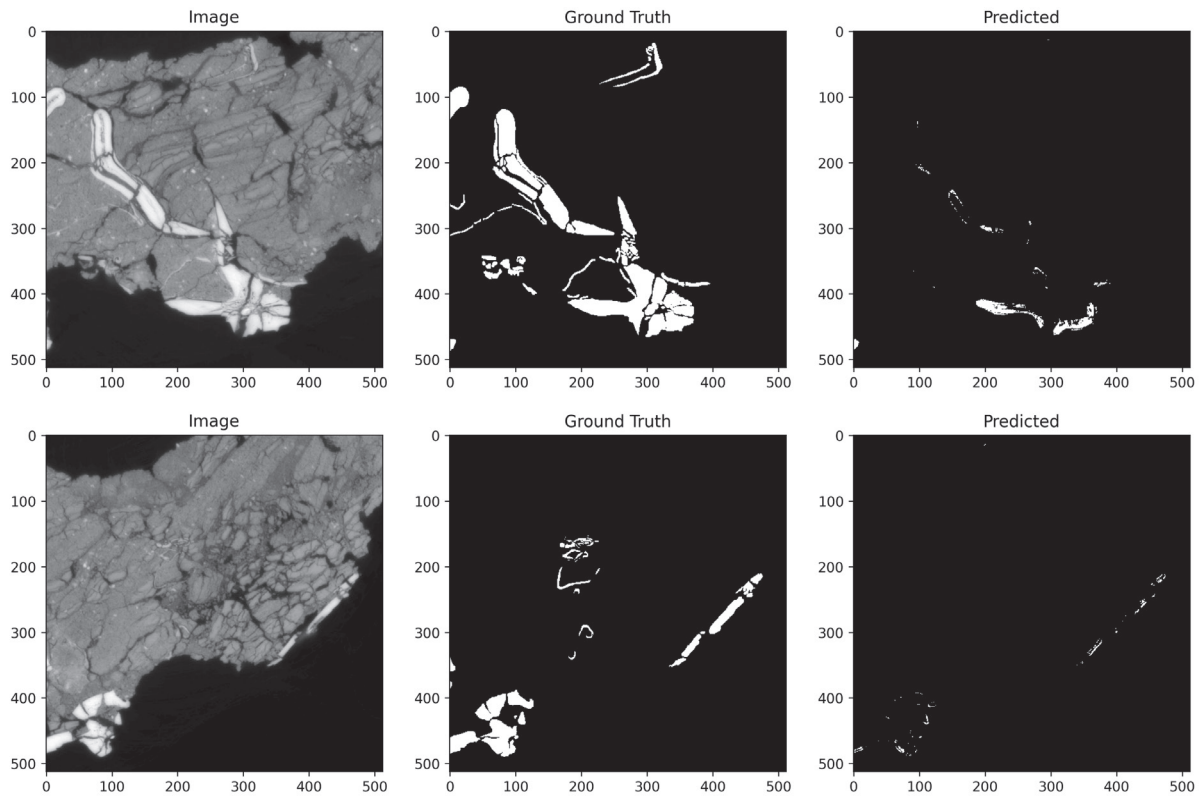


Figure 8. Paddlefish $\sim 45 \mu\text{m}$ segmentation samples. The model tended to confuse bone for other high-brightness objects in the scene, likely due to underparametrisation and an inability to distinguish bone from other bright sources using the necessary 3D context.

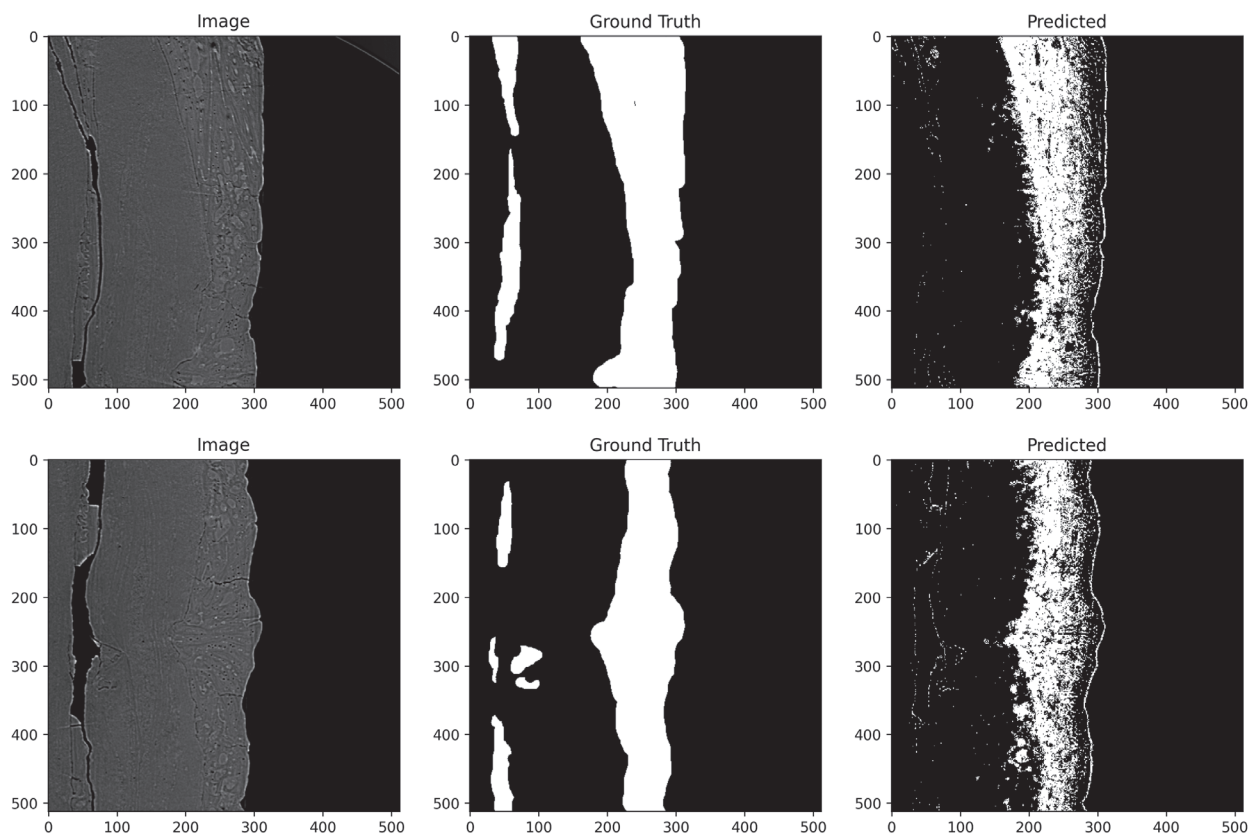


Figure 9. Nothosaur segmentation samples. Two samples of the Nothosaur dataset, illustrating that the texture of the imagery (left panels) was beyond the capabilities of a simple random forest segmentation model.

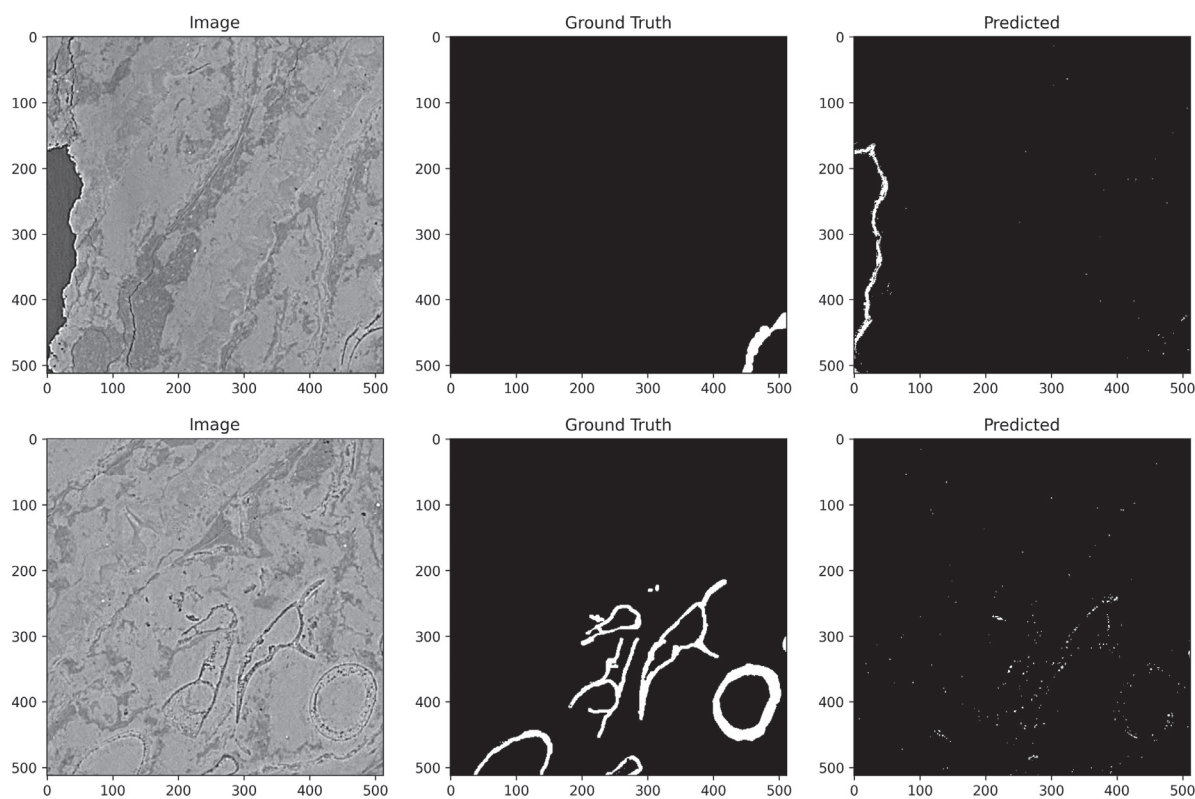


Figure 10. Phu Phok Egg segmentation samples. This challenging dataset illustrates a major failure of the default random forest segmentation model with the multiscale basic pixelwise features in 2D. The high phase contrast in the scan likely caused the model to excessively interpret areas of strong phase contrast as bone, leading it to incorrectly identify other high-contrast edges, such as the boundary of the sample, rather than the low-contrast bones.

As noted in the introduction, manual segmentation is labour-intensive, often taking months or years to complete, which delays scientific advancement. The *ml4paleo* tool offers a valuable contribution by democratising access to segmentation technology, making it accessible to research groups that lack the resources for commercial software solutions.

The results of this study demonstrate that the tool performs reliably on datasets with clear contrast and well-defined features, such as the Burrow dataset, where human-level performance was achieved. The high scores in this case indicate that the model effectively identifies and segments relevant structures. The strong performance on the Paddlefish $\sim 13\ \mu\text{m}$ dataset, also highlights the tool's capability to handle high-resolution scans with complex anatomical features with a dramatic labour improvement over human annotation alone. However, the drop in performance on the Paddlefish $\sim 45\ \mu\text{m}$ dataset shows that lower-resolution data pose challenges.

The Nothosaur and Phu Phok Egg datasets further highlight the tool's limitations, particularly when dealing with both low contrast as well as a combination of high phase contrast with complex low-contrast textures. The segmentation models struggled significantly on both datasets. In the case of the Nothosaur dataset, it is worth noting that obtaining a manual segmentation required over six weeks of full-time effort from a human expert. This underscores the extreme difficulty of the task and puts our tool's poor performance into perspective, as even manual segmentation was highly labour-intensive due to poor specimen quality. Similarly, segmenting the Phu Phok Egg dataset took approximately eight weeks (pers. comm. Vincent Fernandez), while the Paddlefish scans were segmented in two weeks, of which the majority of time was spent on the $\sim 45\ \mu\text{m}$ dataset. The burrow dataset, however, was segmented quite quickly using simple pixel value thresholding, a process that involves selecting an intensity value and classifying pixels below this threshold as false and those above it as true. All manual segmentations we compared to were performed in VGStudio MAX (Volume Graphics, Heidelberg, Germany) and with substantial computing power costs.

These results highlight the need for further refinement of the tool and addition of more sophisticated segmentation models, particularly in its ability to handle more challenging datasets with low contrast and intricate textures.

The performance variations across datasets suggest that future versions of the tool could also benefit from specialised models tailored to different fossil types, resolutions, and scan characteristics. Although the current default segmentation algorithm has limitations, the tool's flexible design allows for the future integration of advanced techniques, such as multi-class segmentation or 3D modelling, to improve its handling of complex fossil data and enhance accuracy across diverse datasets. Although the current default segmentation algorithm has limitations, the tool's flexible design allows for the future integration of advanced techniques, such as multi-class segmentation or 3D modelling, to improve its handling of complex

fossil data and enhance accuracy across diverse datasets. More broadly, the *ml4paleo* tool is intended to make fossil segmentation more efficient and accessible. However, segmentation quality can vary across datasets. To mitigate this, the tool supports the replacement or updating of segmentation algorithms, offering users flexibility and enabling ongoing improvements for more consistent results. Additionally, there is potential to enhance the tool's ability to distinguish closely related structures (e.g., bone and matrix) by incorporating multi-class segmentation and deep learning methods. For example, a deep-learning-based segmentation model with impressive accuracy for fossil CT-scan datasets was recently published by Knutsen and Konovalov (2024). This and similar future models could potentially be integrated into the *ml4paleo* tool. More broadly, the *ml4paleo* tool is intended to make fossil segmentation more efficient and accessible.

We intend for this work to serve as a launch point for future work in progressing the accessibility of palaeontology data and analysis, and that this open-source ecosystem can continue to advance the democratization of science in palaeontology and beyond.

Data & Code availability

All our code is open-source and licenced under the Apache 2.0 licence. The code can be accessed at <https://github.com/j6k4m8/ml4paleo>. All datasets used in this paper are available through the ESRF Palaeontology database (paleo.esrf.eu). A public version of this tool is available at ml4paleo.com.

Acknowledgements

We gratefully acknowledge our colleagues for their valuable input in shaping the functional requirements of the tool. We also extend our thanks to Vincent Fernandez for providing his segmentation data and for sharing information regarding the time required to complete these segmentations. M.A.D.D. and P.E.A. were supported by the Swedish Research Council (VR) under grant 2020-03685. F.K.G. and T.B.S. were supported by Kjell och Märta Beijer Foundation. D.F.A.E. was supported by the Dutch Research Council (NWO) under grant 333.22.013. We would also like to thank Dave Marshall for his podcast *Palaeocast*, through which J.K.M. discovered the work of M.A.D.D. and colleagues, leading to the initiation of this collaboration. Reviewer Donald Henderson and editor Alexander Schmidt are thanked for their helpful feedback and suggestions.

References

- Bishop C, Matelsky J, Wilt M, Downs J, Rivlin P, Plaza S, Wester B, Gray-Roncal W (2021) Confirms: A toolkit for scalable, black box connectome assessment and investigation. In: Proceedings of the

- 2021 43rd Annual International Conference of the IEEE Engineering in Medicine & Biology Society (EMBC). IEEE, 2444–2450. <https://doi.org/10.1109/EMBC46164.2021.9630109>
- Charles AS, Falk B, Turner N, Pereira TD, Tward D, Pedigo BD, Chung J, Burns R, Ghosh SS, Kebschull JM, Silversmith W, Vogelstein JT (2020) Toward community-driven big open brain science: open big data and tools for structure, function, and genetics. *Annu Rev Neurosci*. 43(1): 441–464. <https://doi.org/10.1146/annurev-neuro-100119-110036>
- During MAD, Smit J, Voeten DFEA, Berruyer C, Tafforeau P, Sanchez S, Stein KHW, Verdegaaal-Warmerdam SJA, van der Lubbe JHJL (2022) The Mesozoic terminated in boreal spring. *Nature* 603(7899): 91–94. <https://doi.org/10.1038/s41586-022-04446-1>
- Edie SM, Collins KS, Jablonski D (2023) High-throughput micro-CT scanning and deep learning segmentation workflow for analyses of shelly invertebrates and their fossils: Examples from marine bivalvia. *Frontiers in Ecology and Evolution* 11: 1127756. <https://doi.org/10.3389/fevo.2023.1127756>
- European Synchrotron Radiation Facility (n.d.) ESRF Paleontology Database [Internet]. <http://paleo.esrf.eu>
- Fernandez V, Abdala F, Carlson KJ, Cook DC, Rubidge BS, Yates A, Tafforeau P (2013) Synchrotron reveals early Triassic odd couple: injured amphibian and aestivating therapsid share burrow. *PLoS ONE* 8(6): e64978. <https://doi.org/10.1371/journal.pone.0064978>
- Fernandez V, Buffetaut E, Suteethorn V, Rage JC, Tafforeau P, Kundrat M (2015) Evidence of egg diversity in squamate evolution from Cretaceous anguimorph embryos. *PLoS ONE* 10(7): e0128610. <https://doi.org/10.1371/journal.pone.0128610>
- Grinberg M (2018) Flask web development: developing web applications with Python. Sebastopol: O'Reilly Media, Inc.
- Harris CR, Millman KJ, van der Walt SJ, Gommers R, Virtanen P, Cournapeau D, Wieser E, Taylor J, Berg S, Smith NJ, Kern R, Picus M, Hoyer S, van Kerkwijk MH, Brett M, Haldane A, Del Río JF, Wiebe M, Peterson P, Gérard-Marchant P, Sheppard K, Reddy T, Weckesser W, Abbasi H, Gohlke C, Oliphant TE (2020) Array programming with NumPy. *Nature* 585: 357–362. <https://doi.org/10.1038/s41586-020-2649-2>
- Hider Jr R, Kleissas D, Gion T, Xenos D, Matelsky J, Pryor D, Rodriguez L, Johnson EC, Gray-Roncal W, Wester B (2022) The brain observatory storage service and database (bosssdb): A cloud-native approach for petascale neuroscience discovery. *Front Neuroinform* 16: 828787. <https://doi.org/10.3389/fninf.2022.828787>
- Hilton EJ, During MAD, Grande L, Ahlberg PE (2023) New paddlefishes (acipenseriformes, polyodontidae) from the late Cretaceous Tanis site of the Hell Creek Formation in North Dakota, USA. *Journal of Paleontology* 97(3): 1–18. <https://doi.org/10.1017/jpa.2023.19>
- Knutsen EM, Konovalov DA (2024) Accelerating segmentation of fossil CT scans through Deep Learning. *Scientific Reports* 14: 20943. <https://doi.org/10.1038/s41598-024-71245-1>
- Matelsky JK, Rodriguez LM, Xenos D, Gion T, Hider R, Wester BA, Gray-Roncal W (2021) An integrated toolkit for extensible and reproducible neuroscience. In: Proceedings of the 2021 43rd Annual International Conference of the IEEE Engineering in Medicine & Biology Society (EMBC). IEEE, 2413–2418. <https://doi.org/10.1109/EMBC46164.2021.9630199>
- Miles A, Kirkham J, Durant M, Bourbeau J, Onalan T, Hamman J (2020) zarr-developers/zarr-python: v2.4.0 [Internet]. <https://github.com/zarr-developers/zarr-python>
- Pedregosa F, Varoquaux G, Gramfort A, Michel V, Thirion B, Grisel O, et al. (2011) Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research* 12: 2825–2830.
- Ronneberger O, Fischer P, Brox T (2015) U-net: Convolutional networks for biomedical image segmentation. In: Navab N, Hornegger J, Wells WM, Frangi AF (Eds) Medical Image Computing and Computer-Assisted Intervention–MICCAI 2015: 18th International Conference. Munich, Germany: Springer, 234–241. https://doi.org/10.1007/978-3-319-24574-4_28
- Tafforeau P, Boistel R, Boller E, Bravin A, Brunet M, Chaimanee Y, Cloetens P, Feist M, Horszowska J, Jaeger J-J, Kay RF, Lazzari V, Marivaux L, Nel A, Nemoz C, Thibault X, Vignaud P, Zabler S (2006) Applications of x-ray synchrotron microtomography for non-destructive 3D studies of paleontological specimens. *Applied Physics A* 83: 195–202. <https://doi.org/10.1007/s00339-006-3507-2>
- Tanti M, Berruyer C, Tafforeau P, Muscat A, Farrugia R, Scerri K, Valentino G, Solé VA, Briffa JA (2021) Automated segmentation of microtomography imaging of Egyptian mummies. *PLoS ONE* 16(12): e0260707. <https://doi.org/10.1371/journal.pone.0260707>
- Voeten DFEA, Reich T, Araujo R, Scheyer TM (2018) Synchrotron microtomography of a nothosaurus marchicus skull informs on nothosaurian physiology and neurosensory adaptations in early sauropterygia. *PLoS ONE* 13(1): e0188509. <https://doi.org/10.1371/journal.pone.0188509>