Convpaint - Universal framework for interactive pixel classification using pretrained neural networks

Lucien Hinderling [1,2,✉], Guillaume Witz [3], Roman Schwob [1], Ana Stojiljkovic [3], Maciej Dobrzyński [1], Mykhailo Vladymyrov [3], Joël Frei [1], Benjamin Grädel [1,2], Agne Frismantiene [1], and Olivier Pertz [1,✉]

[1]Institute of Cell Biology, University of Bern, Baltzerstrasse 4, 3012 Bern, Switzerland
[2]Graduate School for Cellular and Biomedical Sciences, University of Bern, Baltzerstrasse 4, 3012 Bern, Switzerland
[3]Data Science Lab, University of Bern, Sidlerstrasse 5, 3012 Bern, Switzerland

We develop Convpaint, a universal computational framework for interactive pixel classification. Convpaint utilizes pre-trained convolutional neural networks (CNNs) or vision transformers (ViTs) for feature extraction and enables facile segmentation across a wide variety of tasks. Available within the Python-based Napari ecosystem, Convpaint integrates seamlessly with other plugins into image processing pipelines, which we demonstrate with four workflows across different data modalities.

image analysis | pixel classification | multi-dimensional data

Correspondence: ***lucien.hinderling@unibe.ch, olivier.pertz@unibe.ch***

Many bioimage analysis pipelines start with a segmentation step. While deep learning methods (DL) offer high classification accuracy, they require extensive annotated ground truth datasets and dedicated hardware for training. Even foundational models, trained on more diverse data and expected to generalize to new applications without retraining, still need retraining for basic research purposes in practice [1–3]. In contrast, machine learning (ML) approaches using small models that can be trained interactively with sparse annotations have proven to be highly effective (Ilastik, Trainable Weka, Qupath) [4–6]. These approaches traditionally relied on hand-crafted filterbanks to extract image features and train an ML model from sparse annotations and corresponding features, to predict the class of each pixel in the rest of the image or new images. While these models are quick to train and hand-crafted filterbanks effectively describe texture or local image structures [7], breakthrough performance in capturing semantically meaningful information from images has been achieved through learning features, specifically convolutional filters in DL models [8]. Convpaint builds on this work, striking a balance between training speed, accuracy, and steerability by combining ML models that are fast to train with the power of DL. Instead of training a DL model from scratch, Convpaint extracts features from selected layers of pre-trained convolutional neural networks (CNN) or vision transformers (ViT) and uses them to train a random forest classifier. Convpaint is designed in a modular fashion, allowing the feature extractor to be easily replaced by any algorithm that returns local features from an input image (Fig. 1A).

Implemented as a graphical user interface within the Napari ecosystem, Convpaint offers a user-friendly tool for researchers to repurpose pre-trained DL models for their specific tasks without requiring coding or ML expertise (Fig. 1B). Unlike neural networks trained to detect specific structures (e.g., spots, fibers), users can guide Convpaint's output by labeling different regions of interest with scribbles (Fig. 1C). This interactive process, which can be completed in seconds, allows for iterative cycles of annotation and evaluation, rapidly improving the quality of segmentation results. Convpaint seamlessly handles multi-dimensional data, making it suitable for segmenting time-series and 3D data (Fig. 1D,E). It can be trained on an arbitrary number of input channels and output classes (Fig. 1F, shown on synthetic data in S1). When coupled with different feature extraction models, Convpaint can be applied to bioimages across different scales, from subcellular to cellular structures to animals (Fig. 1G, S2). For experienced users, Python APIs are available, allowing them to programmatically control Convpaint. The software is interoperable with a wide range of Napari plugins, enabling complex image analysis workflows within a single software ecosystem without coding. We demonstrate this in three workflows.

Workflow 1 highlights Convpaint's capability to work with multichannel data. Imaging mass cytometry (IMC), spatial transcriptomics, or multiplexed immunofluorescence imaging, can image numerous biomarkers in the same sample. This provides a wealth of information that presents new challenges for data analysis, particularly for interactive data exploration. Fig. 2A-D shows an exemplary use case on a 43-channel IMC dataset [9]. The data can be interactively loaded and browsed using napari-imc [10]. Instead of exporting the data for pixel classification in external software like Ilastik as demonstrated in a previous study [11], pixel classification can be performed directly in Napari using Convpaint. Here, we segment vein and tissue regions and identify markers that are differentially expressed between the two classes.

Workflow 2 exemplifies analysis of a timelapse dataset, of which interactive visualization is natively supported in Napari, and supports training by annotations on multiple

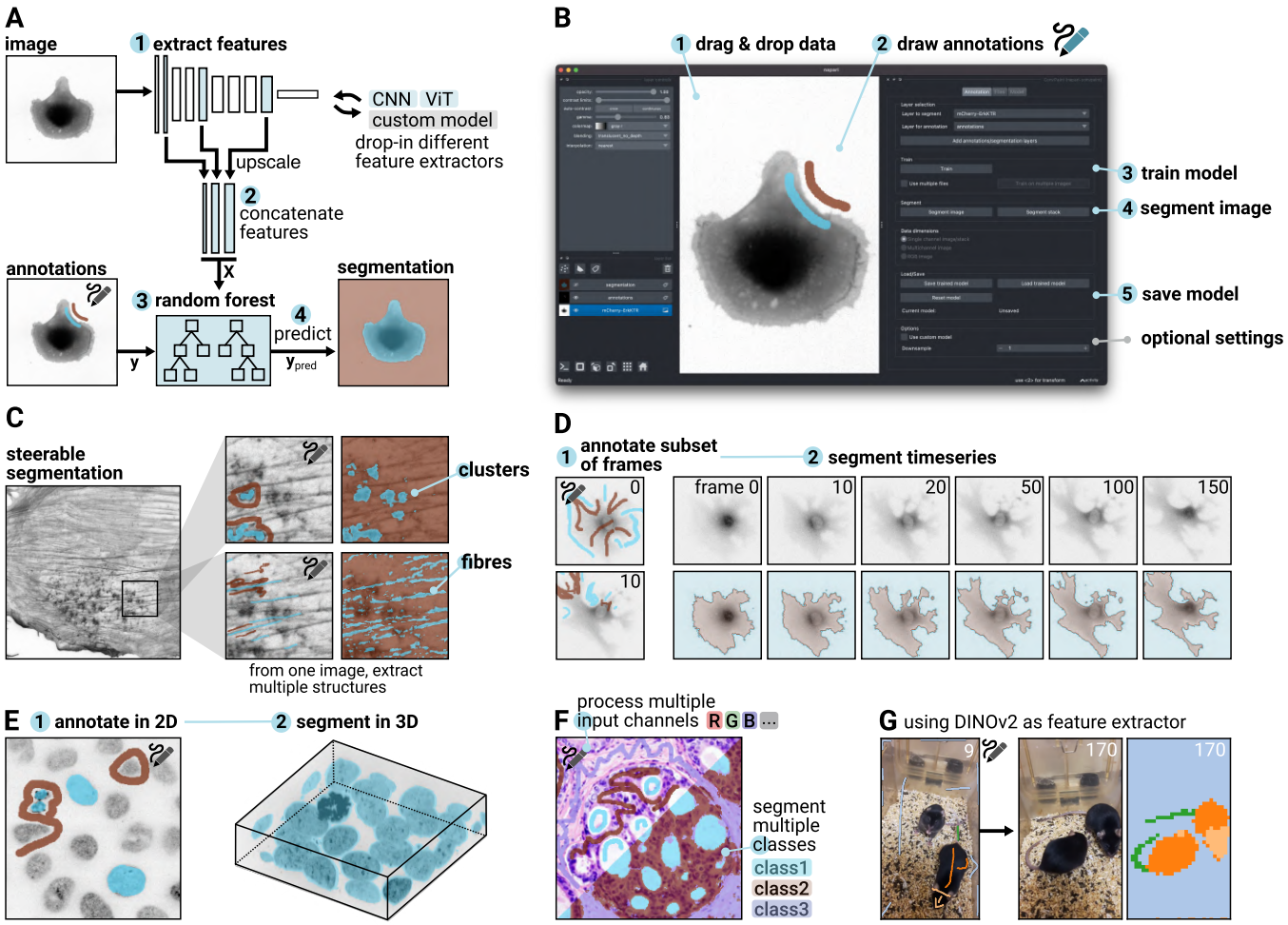

Fig. 1. Overview of the Convpaint algorithm, user interface, and capabilities. **A** Convpaint architecture: 1) Features are extracted from multiple scalings of the input image using a pretrained neural network, 2) upscaled, and concatenated. 3) A random forest model, trained on sparse annotations, predicts the class for each pixel. Different feature extractor models can be used. **B** User interface in Napari: 1) Supports various input formats. 2) Annotations drawn using the labels layer. 3) Single-click model training. 4) Single-click image segmentation with results displayed in a labels layer. 5) Model saving for future use. 6) Advanced settings for custom models and normalization. **C** Interactive adjustment of extracted structures based on annotations. **D** Segments time-series data across all frames with a single click, enabling immediate playback **E** Use Napari's visualization to verify 3D segmentation results.. **F** Adapts to any number of input channels and output classes. **G** DINOv2 as a feature extractor allows the segmentation of macroscopic objects and scenes. Full data shown in fig. S5, supplementary movie M3.

frames. Convpaint is used to detect collective calcium signaling waves in an epithelial monolayer expressing a calcium biosensor (Fig. 2E-I, supplementary movie M1). Convpaint is used to segment cells, napari-assistant [12] is used for post-processing the labels and extracting biosensor information, and ARCOS [13] is utilized to detect and quantify collective signaling events.

To fully exploit information in 3D datasets using Convpaint, we implemented feature extraction from xy, xz, yz projections. The features are then concatenated to form the input for the random forest classifier. In **Workflow 3**, fig. 2J-L, this approach's effectiveness is demonstrated on a light-sheet timelapse dataset of 3D mammary acini expressing a nuclear marker and an ERK biosensor. Following Convpaint segmentation, a simple 3D watershed is used for instance segmentation, and nuclei are tracked with an overlap-based algorithm (Supplementary movie M2). Fig. S3 details how segmentation can be used to extract single-cell ERK activity signaling trajectories. Fig. S4 shows how the use of information from multiple projections improves segmentation performance on a synthetic 3D dataset.

Finally we demonstrate that using ViTs as feature extractors enables Convpaint to track animal body parts and detect behavior—tasks that were previously unattainable with pixel classifiers. For example, DINOv2 that is pre-trained on a large diverse image dataset [14], allows for the segmentation of the head and tail of mice (Fig. S5, supplementary movie M3) and the detection of whether eyes are open or closed (Fig. S6, M4), which is valuable for behavioral and ecological studies. Remarkably, this approach is broadly applicable across different animal species. It can track body parts, such as those of sharks (Fig. S7, M5), with minimal annotations, even when the shark rotates around its axis.

To quantitatively evaluate Convpaint's segmentation performance and compare different feature extractors, we developed a computational pipeline that automatically generates human-like scribbles for existing segmentation datasets with available ground truth annotations. We found significant segmentation performance improvements on

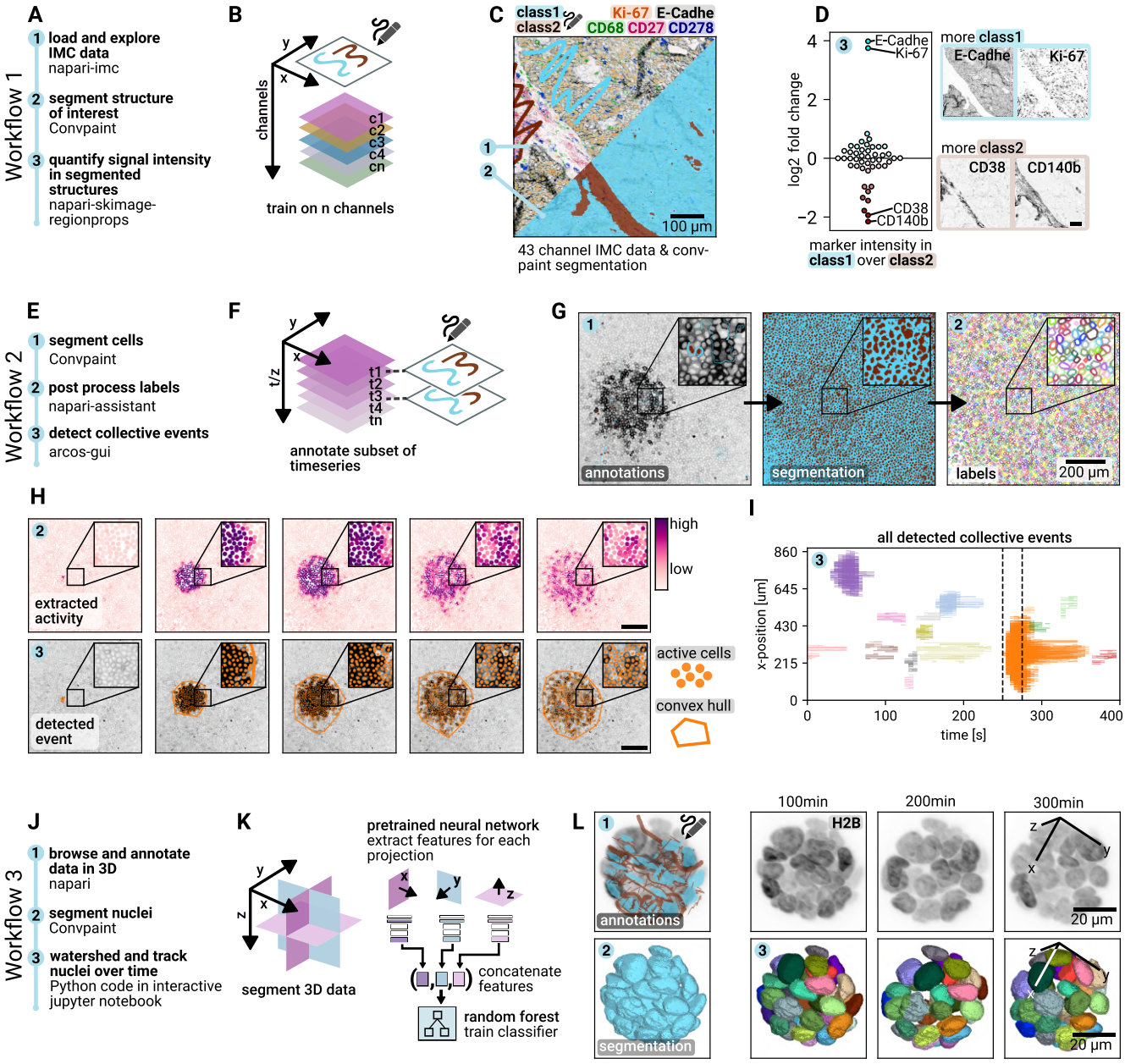

Fig. 2. Image analysis workflows using Convpaint. Workflow 1 (multichannel dataset): **A** Example with multichannel IMC data. **B** Handles arbitrary input channels. **C** Interactive exploration with napari-imc. Labeled structures guide segmentation across all channels. Scale bar: 100 µm. **D** Use class labels for data exploration and statistical analysis, such as identifying differentially expressed markers. Scale bar: 100 µm. **Workflow 2 (time-series):** Supports 3D and time-series data. **E** Combined with arcos-gui to detect collective signaling events in MDCK cell movies. **F** Train classifier on multiple frames or z-slices to predict the rest. **G** Segmentation and post-processing with napari-assistant. Scale bar: 20 µm. **H** Example of collective event detection over 5 frames, with signaling activity and event overlay. Scale bar: 200 µm. **I** Overview of all detected events, with the period from panel H marked. **Workflow 3 (3D segmentation)**: **J** Segmentation of MCF10A acini from lightsheet microscopy, with 3D watershed instance segmentation and tracking with trackpy. **K** Feature extraction from multiple projections (xy, xz, yz) combined for random forest classification. **L** 3D rendering of tracked nuclei with color-coded IDs. Scale bar: 20 µm.

complex data, such as detecting cancerous tissues in histology slides. Results are discussed in detail supplementary information and in fig. S8. Randomly sampled results for different datasets are shown in figs. S9,S10,S11.

The results exemplify Convpaint's flexibility and performance, which, together with its seamless integration within the Napari ecosystem, makes it an attractive initial segmentation step for a wide variety of image analysis tasks.

The code is open source (BSD-3) and installable from the Napari plugin hub[1]. It runs on all common operating systems and standard consumer hardware. Installation instructions, documentation, and video tutorials are available[2].

[1] https://www.napari-hub.org/plugins/napari-Convpaint
[2] https://guiwitz.github.io/napari-Convpaint/book/Landing.html

ACKNOWLEDGEMENTS

This work has been supported by the Chan Zuckerberg Initiative (CZI) grant NP2-0000000095 to LH and OP, Uniscientia fellowship 187-2021 to OP, and Schweizerischer Nationalfonds (SNF) grant 310030_185376 to OP. We thank Scientific Center for Optical and Electron Microscopy (ScopeM) of ETH Zurich, Switzerland, for access to their instruments and services and Dr. Tobias Schwartz for his assistance in acquiring lightsheet data. Calcium imaging data was kindly provided by Yasuto Takeuchi and Yasuyuki Fujita. Other microscopy experiments were performed on equipment supported by the Microscopy Imaging Center (MIC), University of Bern, Switzerland. The mouse icon in A by DBCLS `https://togotv.dbcls.jp/en/pics.html` is CC-BY 4.0 licensed.

AUTHOR CONTRIBUTIONS

LH conceptualized the work. LH, GW, RS, AS, MD, MV, and BG contributed to the development of the software and documentation. RS quantified performance. JF, LH, BG, RS, and AF acquired data. Figures were created by LH. LH and OP wrote the manuscript and acquired funding. All authors read and approved the final manuscript.

COMPETING FINANCIAL INTERESTS

The authors declare no competing financial interests.

Bibliography

[1] Valentin Koch, Sophia J. Wagner, Salome Kazeminia, Ece Sancar, Matthias Hehr, Julia Schnabel, Tingying Peng, and Carsten Marr. Dinobloom: A foundation model for generalizable cell embeddings in hematology, 2024.

[2] Richard J Chen, Tong Ding, Ming Y Lu, Drew FK Williamson, Guillaume Jaume, Bowen Chen, Andrew Zhang, Daniel Shao, Andrew H Song, Muhammad Shaban, et al. Towards a general-purpose foundation model for computational pathology. *Nature Medicine*, 2024.

[3] Ramon Pfaendler, Jacob Hanimann, Sohyon Lee, and Berend Snijder. Self-supervised vision transformers accurately decode cellular state heterogeneity. January 2023. doi:10.1101/2023.01.16.524226.

[4] Stuart Berg, Dominik Kutra, Thorben Kroeger, Christoph N. Straehle, Bernhard X. Kausler, Carsten Haubold, Martin Schiegg, Janez Ales, Thorsten Beier, Markus Rudy, Kemal Eren, Jaime I. Cervantes, Buote Xu, Fynn Beuttenmueller, Adrian Wolny, Chong Zhang, Ullrich Koethe, Fred A. Hamprecht, and Anna Kreshuk. ilastik: interactive machine learning for (bio)image analysis. *Nature Methods*, September 2019. ISSN 1548-7105. doi:10.1038/s41592-019-0582-9.

[5] Ignacio Arganda-Carreras, Verena Kaynig, Curtis Rueden, Kevin W Eliceiri, Johannes Schindelin, Albert Cardona, and H Sebastian Seung. Trainable Weka Segmentation: a machine learning tool for microscopy pixel classification. *Bioinformatics*, 33(15):2424–2426, 03 2017. ISSN 1367-4803. doi:10.1093/bioinformatics/btx180.

[6] Peter Bankhead, Maurice B Loughrey, José A Fernández, Yvonne Dombrowski, Darragh G McArt, Philip D Dunne, Stephen McQuaid, Ronan T Gray, Liam J Murray, Helen G Coleman, Jacqueline A James, Manuel Salto-Tellez, and Peter W Hamilton. QuPath: Open source software for digital pathology image analysis. *Sci. Rep.*, 7(1), December 2017.

[7] Thomas Leung and Jitendra Malik. *International Journal of Computer Vision*, 43(1):29–44, 2001. ISSN 0920-5691. doi:10.1023/a:1011126920638.

[8] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. In F. Pereira, C.J. Burges, L. Bottou, and K.Q. Weinberger, editors, *Advances in Neural Information Processing Systems*, volume 25. Curran Associates, Inc., 2012.

[9] Nils Eling and Jonas Windhager. Example imaging mass cytometry raw data. February 2022. doi:10.5281/zenodo.5949116.

[10] Jonas Windhager, Bernd Bodenmiller, and Nils Eling. An end-to-end workflow for multiplexed image processing and analysis. *bioRxiv*, 2021. doi:10.1101/2021.11.12.468357.

[11] Jonas Windhager, Vito Riccardo Tomaso Zanotelli, Daniel Schulz, Lasse Meyer, Michelle Daniel, Bernd Bodenmiller, and Nils Eling. An end-to-end workflow for multiplexed image processing and analysis. *Nature Protocols*, 18(11):3565–3613, October 2023. ISSN 1750-2799. doi:10.1038/s41596-023-00881-0.

[12] Robert Haase, Ryan Savill, Peter Sobolewski, and Lee Dohyeon. haesleinhuepf/napari-assistant: 0.4.7, 2023.

[13] Paolo Armando Gagliardi, Benjamin Grädel, Marc-Antoine Jacques, Lucien Hinderling, Pascal Ender, Andrew R. Cohen, Gerald Kastberger, Olivier Pertz, and Maciej Dobrzyński. Automatic detection of spatio-temporal signaling patterns in cell collectives. *Journal of Cell Biology*, 222(10), July 2023. ISSN 1540-8140. doi:10.1083/jcb.202207048.

[14] Maxime Oquab, Timothée Darcet, Théo Moutakanni, Huy Vo, Marc Szafraniec, Vasil Khalidov, Pierre Fernandez, Daniel Haziza, Francisco Massa, Alaaeldin El-Nouby, Mahmoud Assran, Nicolas Ballas, Wojciech Galuba, Russell Howes, Po-Yao Huang, Shang-Wen Li, Ishan Misra, Michael Rabbat, Vasu Sharma, Gabriel Synnaeve, Hu Xu, Hervé Jegou, Julien Mairal, Patrick Labatut, Armand Joulin, and Piotr Bojanowski. Dinov2: Learning robust visual features without supervision. 2024. doi:10.48550/arXiv.2304.07193.

[15] Wilson Bakasa and Serestina Viriri. Vgg16 feature extractor with extreme gradient boost classifier for pancreas cancer prediction. *Journal of Imaging*, 9(7):138, July 2023. ISSN 2313-433X. doi:10.3390/jimaging9070138.

[16] Carsen Stringer and Marius Pachitariu. Transformers do not outperform cellpose. April 2024. doi:10.1101/2024.04.06.587952.

[17] Carsen Stringer, Tim Wang, Michalis Michaelos, and Marius Pachitariu. Cellpose: a generalist algorithm for cellular segmentation. *Nature Methods*, 18(1):100–106, December 2020. ISSN 1548-7105. doi:10.1038/s41592-020-01018-x.

[18] Xiongwei Wu, Xin Fu, Ying Liu, Ee-Peng Lim, Steven C. H. Hoi, and Qianru Sun. A large-scale benchmark for food image segmentation. *CoRR*, abs/2105.05409, 2021.

[19] Mohamed Amgad, Habiba Elfandy, Hagar Hussein, Lamees A Atteya, Mai A T Elsebaie, Lamia S Abo Elnasr, Rokia A Sakr, Hazem S E Salem, Ahmed F Ismail, Anas M Saad, Joumana Ahmed, Maha A T Elsebaie, Mustafijur Rahman, Inas A Ruhban, Nada M Elgazar, Yahya Alagha, Mohamed H Osman, Ahmed M Alhusseiny, Mariam M Khalaf, Abo-Alela F Younes, Ali Abdulkarim, Duaa M Younes, Ahmed M Gadallah, Ahmad M Elkashash, Salma Y Fala, Basma M Zaki, Jonathan Beezley, Deepak R Chittajallu, David Manthey, David A Gutman, and Lee A D Cooper. Structured crowdsourcing enables convolutional segmentation of histology images. *Bioinformatics*, 35(18):3461–3467, February 2019. ISSN 1367-4811. doi:10.1093/bioinformatics/btz083.

[20] Lucien Hinderling, Maciej Dobrzyński, Yasuto Takeuchi, and Olivier Pertz. Calcium waves in mdck epithelium. *BioStudies Database*, 2024. doi:10.6019/s-biad1135.

[21] Pascal Ender, Paolo Armando Gagliardi, Maciej Dobrzyński, Agne Frismantiene, Coralie Dessauges, Thomas Höhener, Marc-Antoine Jacques, Andrew R. Cohen, and Olivier Pertz. Spatiotemporal control of erk pulse frequency coordinates fate decisions during mammary acinar morphogenesis. *Developmental Cell*, 57(18):2153–2167.e6, September 2022. ISSN 1534-5807. doi:10.1016/j.devcel.2022.08.008.

[22] Agne Frismantiene, Lucien Hinderling, and Olivier Pertz. Light sheet 3d timelapse of a human breast cell acini. *BioStudies Database*, 2024. doi:10.6019/s-biad1134.

[23] Stéfan van der Walt, Johannes L. Schönberger, Juan Nunez-Iglesias, François Boulogne, Joshua D. Warner, Neil Yager, Emmanuelle Gouillart, and Tony Yu. scikit-image: image processing in python. *PeerJ*, 2:e453, June 2014. ISSN 2167-8359. doi:10.7717/peerj.453.

[24] Dmitry V. Sorokin, Igor Peterlik, Vladimir Ulman, David Svoboda, Tereza Necasova, Katsiarina Morgaenko, Livia Eiselleova, Lenka Tesarova, and Martin Maska. Filogen: A model-based generator of synthetic 3-d time-lapse sequences of single motile cells with growing and branching filopodia. *IEEE Transactions on Medical Imaging*, 37(12):2630–2641, December 2018. ISSN 1558-254X. doi:10.1109/tmi.2018.2845884.

[25] Vebjorn Ljosa, Katherine L Sokolnicki, and Anne E Carpenter. Annotated high-throughput microscopy image sets for validation. *Nature Methods*, 9(7):637–637, June 2012. ISSN 1548-7105. doi:10.1038/nmeth.2083.

[26] Romina Burla, Mattia La Torre, Giorgia Zanetti, Alex Bastianelli, Chiara Merigliano, Simona Del Giudice, Alessandro Vercelli, Ferdinando Di Cunto, Marina Boido, Fiammetta Vernì, and Isabella Saggio. p53-sensitive epileptic behavior and inflammation in ft1 hypomorphic mice. *Frontiers in Genetics*, 9, November 2018. ISSN 1664-8021. doi:10.3389/fgene.2018.00581.

Methods

Convpaint implementation details. Convpaint features a modular architecture designed to accommodate a wide range of feature extractors, enhancing existing algorithms or pre-trained models with added steerability. We compare three different types of feature extractors:

CNN We implement the VGG16 [15] architecture, pre-trained on the ImageNet dataset, to extract local image features such as edges, textures, and color channel correlations when working with RGB images. Downscaled versions of the input image are passed through VGG16, creating a featurized image pyramid. These features are then upscaled and concatenated with unscaled outputs and deeper CNN layer features, balancing segmentation speed and accuracy. This method effectively generalizes to a variety of image segmentation tasks (Fig. S2). We evaluated different configurations of input scalings and layers for feature extraction and provide default settings that perform well on all of the tested datasets.

ViT We incorporate two ViT models—DINOv2 [14] and UNI [2]. DINOv2 is pretrained on 142M images from ImageNet, while UNI is pretrained on a large histology dataset. These models extract patch features of 14x14 pixels (DINOv2) and 16x16 pixels (UNI), providing superior performance in certain segmentation tasks despite a loss in resolution for fine details like small cell protrusions. For each patch, 384 features are extracted. For all DINOv2 tests, we used the VIT-S14 distilled model version with registers, as it produced less patch noise in predictions (Figure S8J).

Classical Filterbank To compare the performance of Convpaint when using pre-trained neural networks versus classical filterbanks as feature extractors, we use the filters implemented in napari-ilastik[3]. We chose the maximal combination of filters and sigma parameters suggested in the library, including Gaussian, Laplacian of Gaussian, Gaussian gradient magnitude, difference of Gaussians, structure tensor eigenvalues, and Hessian of Gaussian eigenvalues, with sigma values 0.3, 0.7, 1.0, 1.6, 3.5, 5.0, 10.0. Fig. S12 shows a visual comparison of filters used in classical filterbanks versus learned convolutional filters extracted from VGG16.

Convpaint is optimized for both training and prediction efficiency. It supports:

- Crops Around Annotations: Avoids processing entire images by extracting crops around annotated pixels.
- Tiling and Parallel Processing: Handles large images by tiling them and using parallel processing, with appropriate padding to minimize edge effects. One-click batch processing for image stacks.
- Data Management: Manages larger-than-memory files using Dask, appropriate handling of additional image dimensions (channels vs. time/z-slices)
- Customizability: Users can easily integrate other feature extractors by implementing a simple function that returns a feature matrix from an image. Convpaint takes care of the user interface, classifier training, data management, and parallelization.

For users wanting to implement their own feature extractor, we provide a minimal example that serves as a blueprint for starting development. Convpaint makes it easy to take existing architectures and repurpose them in minutes. To give another example, we have explored using intermediate outputs of a Cellpose model as a feature extractor. While the model is trained to predict cell masks, in Convpaint it can be steered to segment cell boundaries or nuclei with a couple of scribbles. Another possibility is to simply concatenate the output of multiple feature extractors, which allows combining their strengths, e.g., we combined the pixel-accurate segmentation of VGG16 with the semantic understanding of DINOv2 S13. While DINOv2 reliably differentiates shark body-parts, it is limited in accuracy by the patch size of its features. VGG16 on the other hand, precisely masks the shark, but lacks the semantic understanding to correctly label different anatomical structures. Combining both models achieves semantically correct labels with high spatial resolution.

The design of Convpaint streamlines experimentation with different feature extractors and ensures that new innovations in computer vision leverage its performance and usability features, fostering the development of custom feature extractors for specific research needs.

[3] https://github.com/ilastik/ilastik-napari

Quantification of segmentation performance. Assessing Convpaint's performance, especially given its interactive nature, is challenging. Even non-interactive models face problems in unbiased performance evaluation in bioimage analysis [16]. Due to the lack of scribble-annotated datasets, we created an algorithm to generate human-like scribbles for existing ground truth datasets, allowing for an unbiased and quantitative assessment of segmentation performance. For each image from the data set, we generate varying scribble masks with different annotation densities (Fig. S8A). We evaluated three datasets, Cellpose [17], foodseg103 [18], and a subset of a breast cancer histology slide database [19]. We chose the foodseg103 hypothesizing that classical filters would struggle with assigning semantic information for food items with highly variable textures within the same label. Similarly, we selected the breast cancer dataset, which, despite having less diverse images than foodseg103, represents a common use case in biological research. In total we evaluated accuracy on 148k samples. The code to automatically generate scribbles from ground truth, recreate the figures, or explore further results of the analysis is available on github[4].

Scribble generation. To closely mimic human annotations, the scribbles were created by combining three types of algorithmically generated lines:

1. Center ridge lines: Sampled from the primary skeleton of the ground truth mask.
2. Boundary parallel lines: Sampled from the secondary skeleton, which is derived from the ground truth mask after subtracting the primary skeleton.
3. Boundary perpendicular lines: Lines connecting the primary skeleton to the mask boundary.

By varying the sampling density, we can generate different levels of annotation coverage, such as 0.1% or 1% of the image pixels. For the Cellpose dataset, which consists mostly of images with numerous small cell-like objects, we generated a large number of short, 1-pixel-wide scribbles. The ground truth masks were converted from instance segmentation to semantic segmentation (i.e., foreground/background instead of cell IDs). For the foodseg103 dataset, which features fewer but larger regions of different food items, we generated fewer, longer scribbles with a width of 2 pixels. For the histology dataset, we created medium-length scribbles with 2 pixels width.

Dataset evaluation. As quantitative readout, we use mean intersection over union (mIoU). For the Cellpose dataset (540 images, Fig. S8B), which involves segmenting small cell-like structures from a dark background, we observed similar performance between VGG16 filters and classical filter banks (Fig. S8C). Increased performance variability in low annotation regimes was due to the random information content of the annotated pixels. At higher annotation levels, accuracy was often limited by erroneous ground truth annotations, such as missing protrusions (Fig. S8D). DINOv2 performed poorly due to its patch size being too large for small cellular details. Randomly selected sample images and predictions are shown in Figure S9.

To reduce computation time for the foodseg103 dataset (Figure S8E) with 4983 images, we first remove images larger than 640k pixels, then randomly select 520 images of the remaining dataset for evaluation. The images require distinguishing food items with complex textures and colors, VGG16 filters outperformed classical filter banks for all tested configurations. In cases of sparse annotations, smaller networks with VGG16 filters showed better performance, likely due to reduced overfitting. Due to the larger label regions, unlike in the Cellpose dataset, DINOv2's performance was not constrained by patch size and massively outperformed both classical and VGG16 filters, even for very low annotation regimes (Fig. S8F). For all models, we observed diminishing returns in segmentation performance as annotation levels increased. For Convpaint users, this suggests that adding a few more annotations is beneficial when only a minimal number of scribbles are present, but if performance gains plateau, it's a good time to stop (shown for a VGG16 model in Fig. S8G). Randomly selected sample images and predictions are presented in Figure S10.

Motivated by DINOv2's performance on foodseg103, we tested it on a subset of a breast cancer histology slide dataset (Fig. S8H). DINOv2 again clearly outperformed both classical filter banks and VGG16

[4] https://github.com/quasar1357/scribbles_creator

filters. We also evaluated a histology-specific model built with the same architecture as DINOv2 [2], which surprisingly showed no advantage over DINOv2 (Fig. S8I). Predictions from different models (8 out of 10 images tested) are shown in Fig. S11. DINOv2 not only achieved the highest mIOU results but also produced visually less noisy predictions compared to other models. We found that using the updated version of DINOv2 models with registers reduced patch noise in some of the tested images, so we used the register version for all tests (Fig. S8J).

In summary, DINOv2 generalizes well across domains, even outperforming domain-specific models, as seen with the breast cancer histology dataset. However, its larger patch size limits performance on finely detailed images, such as small cellular structures in the Cellpose dataset, where classical filters or VGG16 perform better. VGG16 matches or exceeds classical filters, making it a suitable replacement. While DINOv2 excels in complex tasks with larger label regions, images more easily segmentable by thresholding intensity values, benefit more from classical filters or VGG16.

Methods and data availability workflows.

Workflow 1: IMC multichannel data. IMC data from [9], available on Zenodo[5] were loaded using the napari-imc plugin [10]. Convpaint was trained on one FOV (Fig. 2 shows Patient 01, Panorama 02, Position 1-1). Skimage was used to extract the per channel statistics for the segmented regions. The log2-fold change in signal intensity was calculated using numpy and plotted with matplotlib. Step-by-step instructions for recreating the workflow are available in the documentation.

Workflow 2: MDCK calcium waves. Timelapse data sets of calcium signaling waves were obtained from MDCK epithelial cells that stably express GCaMP6S - a genetically encoded intracellular calcium sensor (imaging data courtesy of Yasuyuki Fujita). The movies were loaded into Napari and segmented using Convpaint. The resulting binary masks were processed with ARCOS [13] to detect and quantify collective signalling events. The code to recreate the figures is available on github `TODO`. We have made the raw imaging data available on the BioImageArchive [20] under the accession number S-BIAD1135. Step-by-step instructions for recreating the workflow are available in the documentation.

Workflow 3: 3D segmentation and nuclei tracking. We demonstrate that Convpaint can effectively segment and track single cells within a dense 3D spheroid. In supplementary Fig. S3, we illustrate how this data can be further processed to extract single-cell ERK signaling activity dynamics. The cells used are MCF10A, expressing a histone H2B nuclear marker and ERK-KTR, which reports ERK activity through reversible nucleus/cytosol translocation following phosphorylation by active ERK [21] (scheme in Fig. S3A). Data were acquired using a lightsheet microscope with a 5-minute time resolution and an isotropic voxel size of 0.145 µm. Raw imaging data and protocols are available on BioImageArchive [22] under accession number S-BIAD1134.

Data availability other figures. The 3D nuclear data in Fig. 1F is part of the scikit-image [23] data module, called *cells3d*, originally provided by the Allen Institute for Cell Science. The synthetic data in Fig. S4 was generated by another group using FiloGen [24] and is available from the Broad Bioimage Benchmark Collection (BBBC046[6]) [25], showing cell PD-ID451/AR1/T024.

The datasets used for performance quantification have all been previously published. Fig. S8B shows the Cellpose dataset [17]; Fig. S8C shows the foodseg103 dataset [18]; and Fig. S11 uses data from the Breast Cancer Semantic Segmentation (BCSS) dataset[7] [19]. Fig. S5 is a supplement[8] to a study on epileptic behavior in mice [26]. Movies in Figs. S6A, B and S7 were acquired by the authors and are available upon request. The movie in Fig. S6D, E is available online[9] for educational purposes under the Mixkit Restricted License. Images in Fig. S2 were acquired by the authors and are available upon

[5] https://doi.org/10.5281/zenodo.5555575
[6] https://bbbc.broadinstitute.org/BBBC046
[7] https://bcsegmentation.grand-challenge.org/BCSS/
[8] https://doi.org/10.3389/fgene.2018.00581.s007
[9] https://mixkit.co/free-stock-video/gray-and-white-rat-32060/

request, except for the histology slide images, which are from Wikimedia Commons[10], or provided by the scikit image library, acquired at the Center for Microscopy And Molecular Imaging (CMMI).

Convpaint code availability. Convpaint is open source (BSD-3), the source code is available on GitHub[11], and can be installed as a Python package from PiPY[12], or via the Napari GUI from the plugin hub[13]. Convpaint runs on all common operating systems and computers with standard consumer hardware, with optional GPU acceleration. Installation instructions, documentation and video tutorials are available[14].

[10] https://commons.wikimedia.org/wiki/File:Breast_DCIS_histopathology_(1).jpg
[11] www.github.com/guiwitz/napari-convpaint
[12] https://pypi.org/project/napari-convpaint/
[13] https://www.napari-hub.org/plugins/napari-convpaint
[14] https://guiwitz.github.io/napari-convpaint/book/Landing.html

Supplementary figures.

- S1: Correlation extraction across multiple color channels.
- S2: Image segmentation across diverse domains.
- S3: Measuring ERK signaling dynamics at the single-cell level in MCF10A acini.
- S4: Improved segmentation performance using multiple projections.
- S5: Detecting mouse body parts in video.
- S6: Eye state detection in humans and mice.
- S7: Detecting shark body parts in video.
- S8: Quantification of segmentation accuracy and model comparison.
- S9: Feature extractor performance on the Cellpose dataset.
- S10: Feature extractor performance on the foodseg103 dataset.
- S11: Feature extractor performance on histology slides.
- S12: Visual comparison of handcrafted vs. learned filters.
- S13: Combining VGG16 and DINOv2 features for enhanced spatial precision at mask boundaries while maintaining semantic information.

Supplementary movies.

- M1: Detection of calcium waves in MCDK cells
- M2: Segmentation of MCF10A acini
- M3: Segmentation of mouse body parts
- M4: Detection of open/closed eyes in humans and mice
- M5: Segmentation of shark body parts

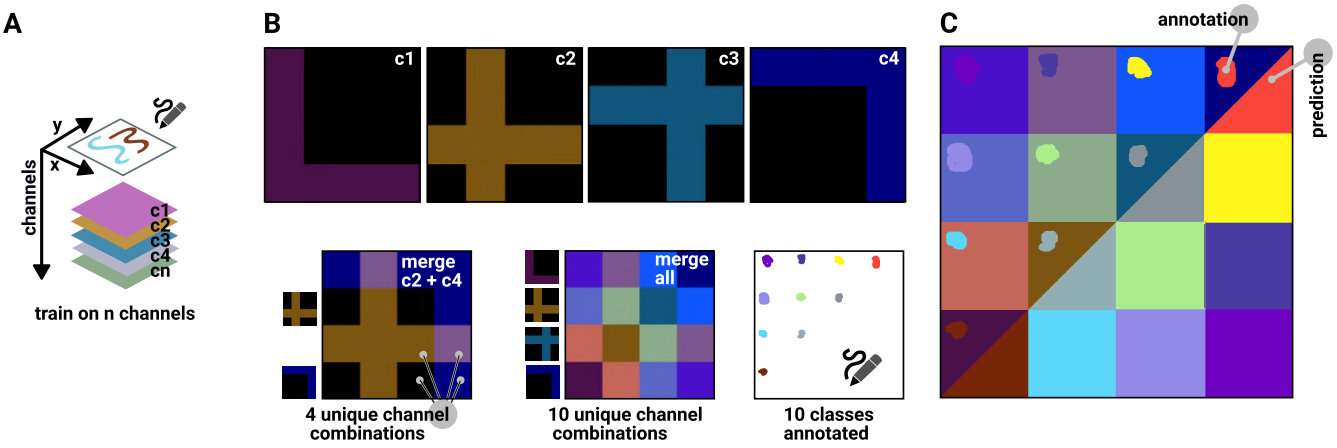

Fig. S1. Correlation extraction across multiple color channels. A The Convpaint classification algorithm can process an arbitrary number of input color channels. **B** This is demonstrated on an artificial image with four channels, which when merged lead to 10 unique color combinations. These 10 combinations are labeled with 10 class labels that can only be reconstructed if the algorithm takes into account the interplay of the different channels. **C** Convpaint correctly predicts the correct class label for pixels that were not labeled, with minor artefacts on boundaries between squares.

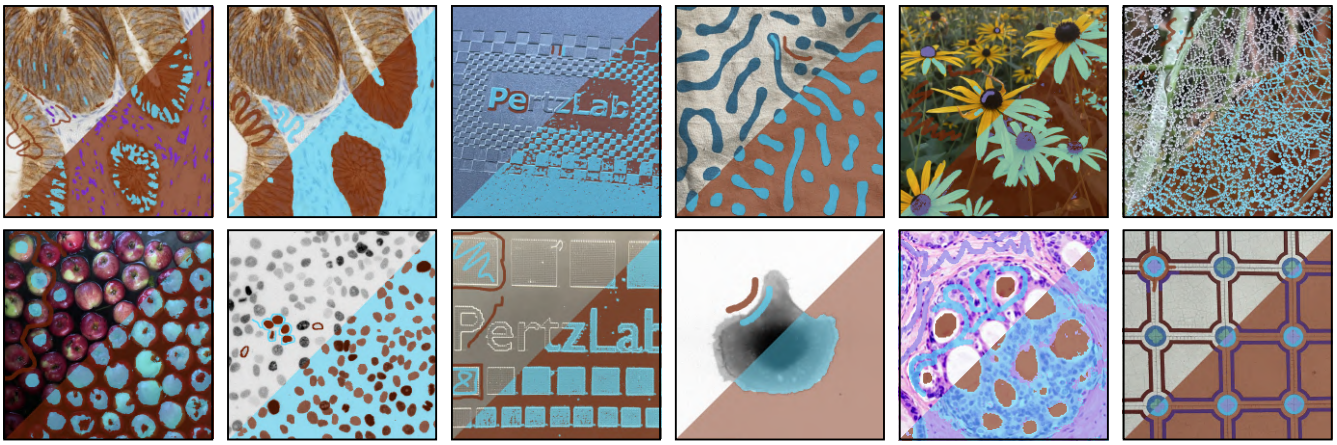

Fig. S2. Image segmentation across diverse domains All images use VGG16 with the default configuration as feature extractor.

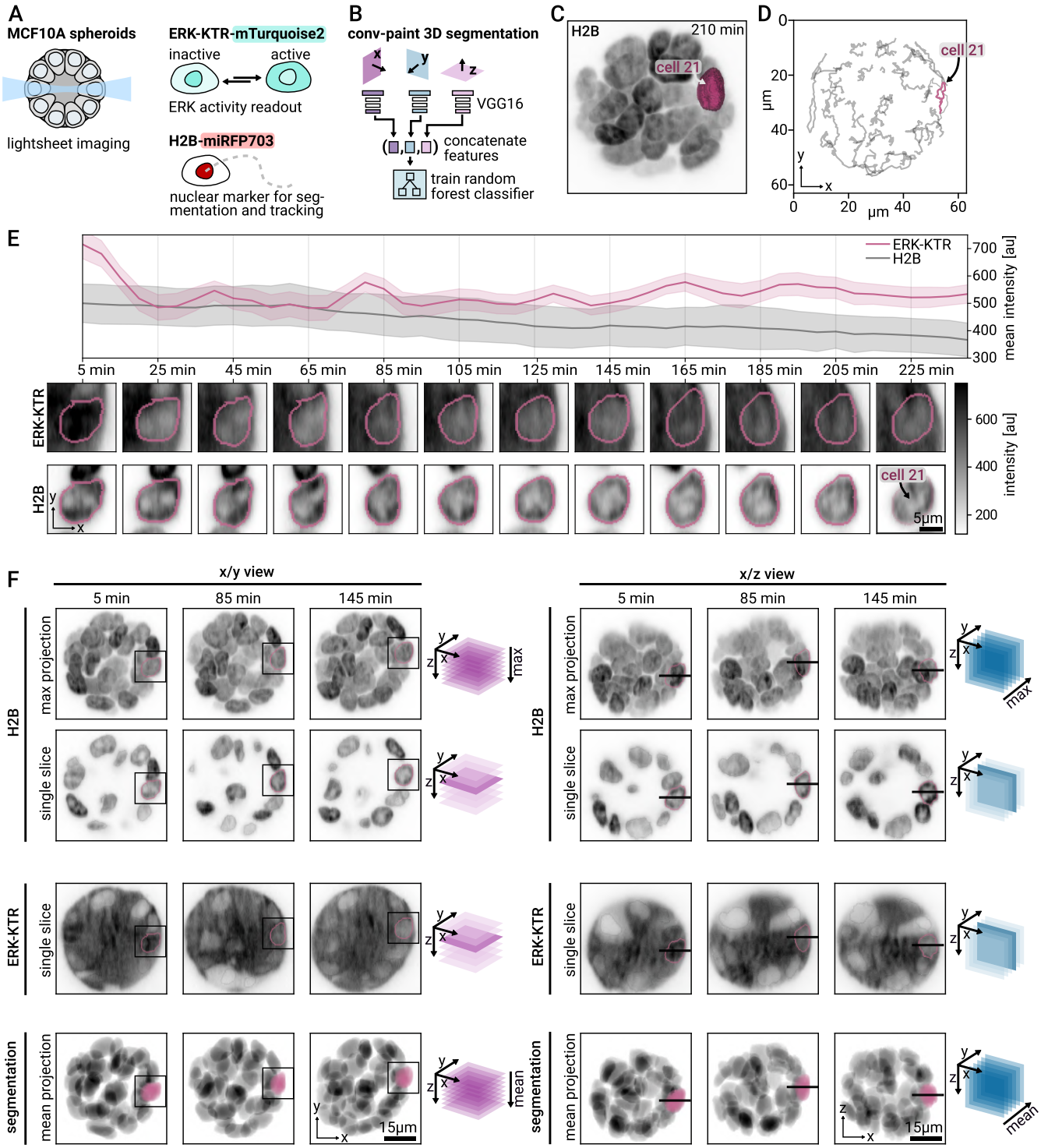

Fig. S3. Measuring ERK signaling dynamics at the single-cell level in MCF10A acini. A Spheroids are imaged with a lightsheet microscope. The cells express an ERK activity sensor and nuclear marker for segmentation and tracking. **B** Convpaint is used to segment the nuclei in 3D. **C** This figure tracks a single cell over time. This panel shows a 3D max projection with the selected cell's mask overlaid. **D** Tracks of all cells from 0 to 250 minutes, selected cell highlighted in color. **E** Mean nuclear ERK-KTR intensity over time as a proxy for ERK activity. We see ERK trajectories as expected and previously described by Ender et al. [21]. In comparison, the mean intensity of the nuclear marker shows some bleaching but no fluctuations otherwise. The images show crops around the selected cell (mean of 3 z-slices, [+1,0,-1] around the z position of the cell centroid). Scalebar is 5 µm. **F** Highlighting the tracked position of the cell within the spheroid for different timepoints, projections, and channels. Box shows insets in panel E. Scalebar is 15 µm.

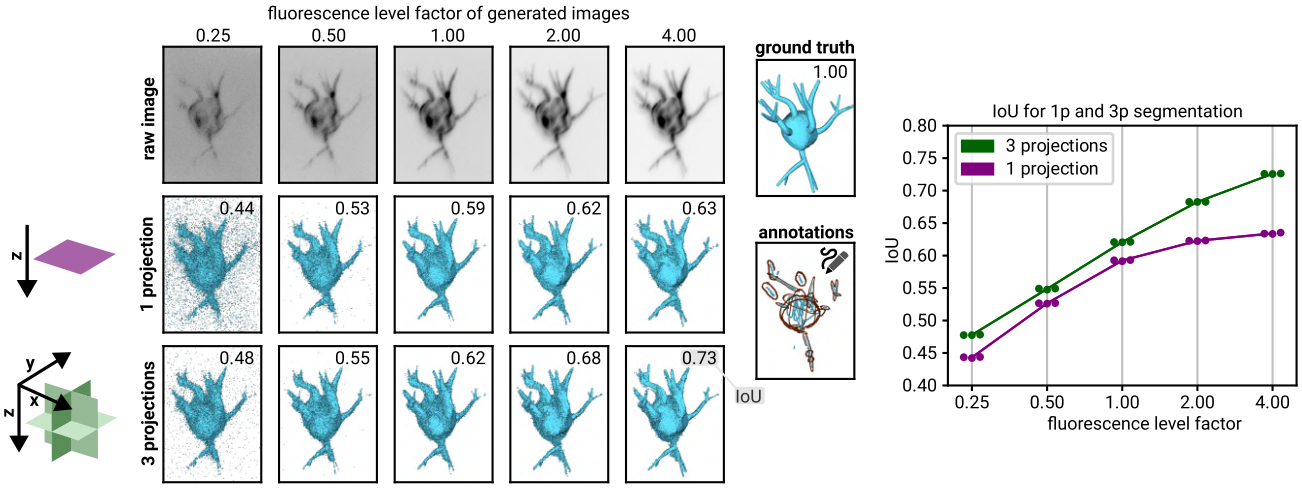

Fig. S4. Improved segmentation performance using multiple projections. Convpaint segmentation performance compared on an artificial cell when extracting features from 1 projection (purple) versus concatenating 3 projections (green), using VGG16 with default configuration as feature extractor. Different signal-to-noise regimes are tested, which are configured by the fluorescence level factor (0.25-4) in the FiloGen software. Performance is measured as intersection over union (IoU). Using 3 projections leads to better segmentation results for all fluorescence level factors. A larger increase in performance is observed for images with a better signal-to-noise ratio.

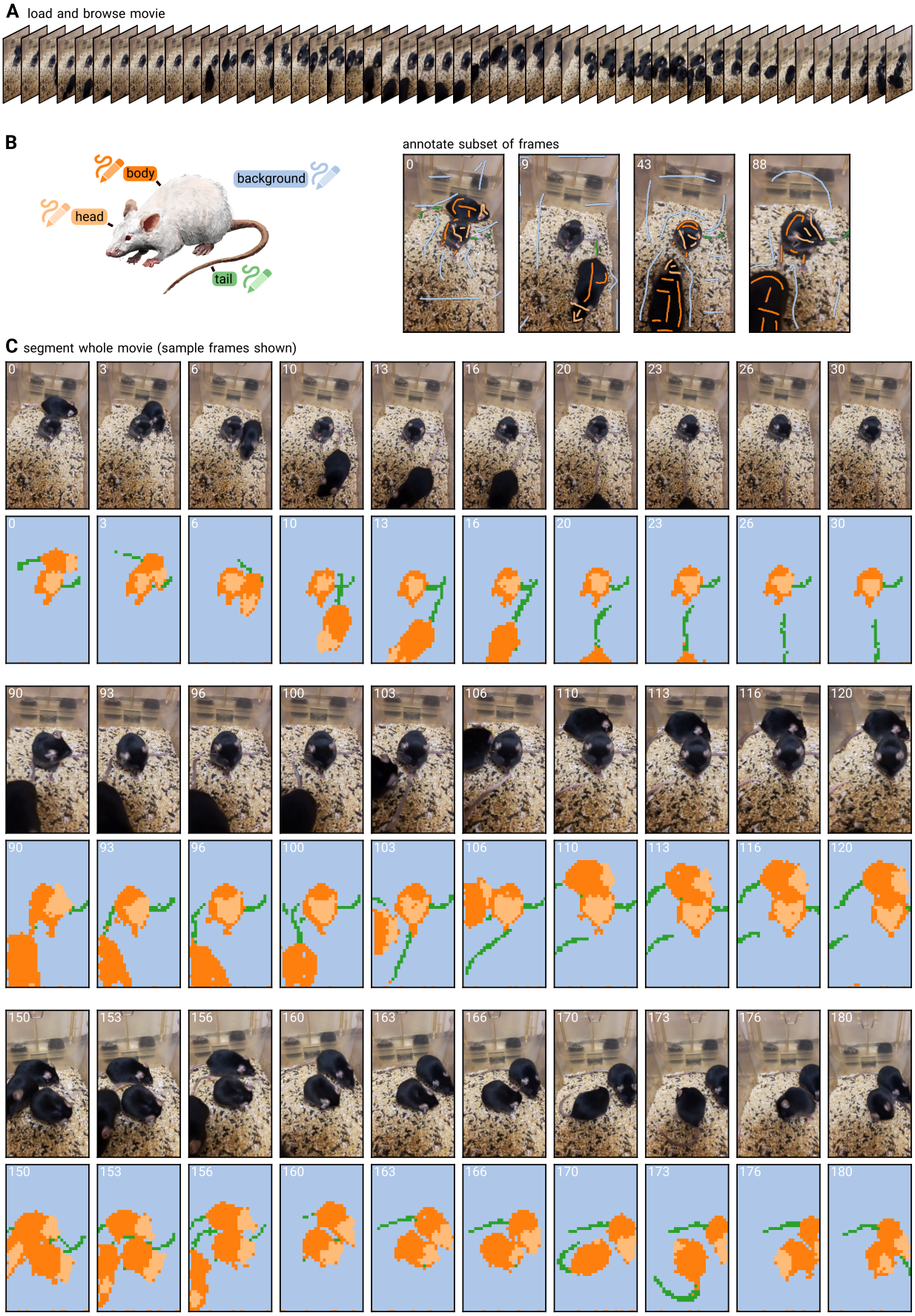

Fig. S5. Detecting mouse body parts in video. A The movie shows mice moving in a cage. The camera is handheld and the mice move in and out of frame. Some frames contain motion blur. **B** Convpaint with DINOv2 as feature extractor is trained on scribbles on four different frames. Head, body, tail, and background are annotated. **C** The trained model is used to predict the rest of the movie (see supplementary movie M3).

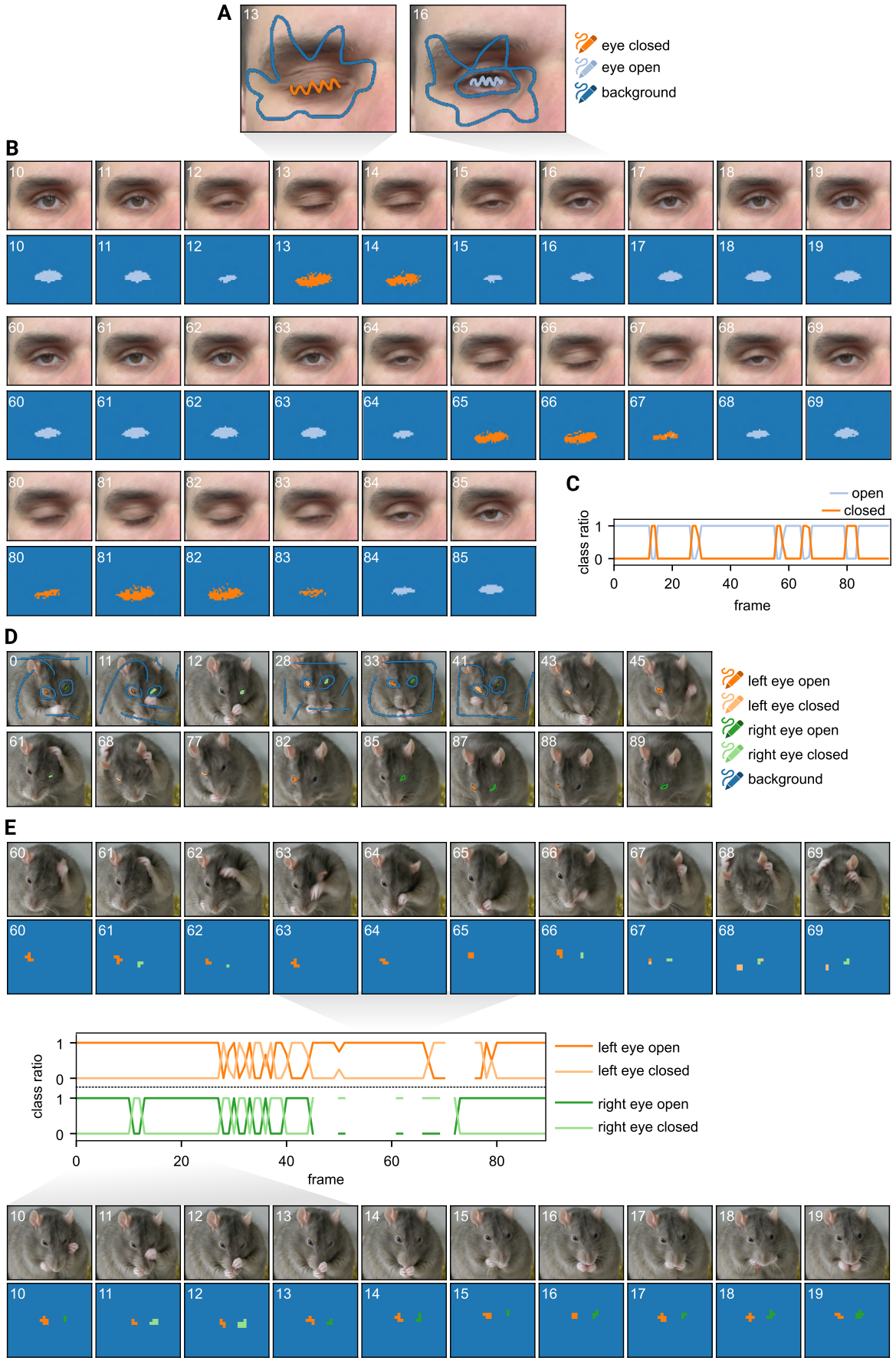

Fig. S6. Eye state detection in humans and mice. A Two frames are labeled with three classes (eye open, eye closed, background) **B** Convpaint is used to segment the rest of the movie. Sample frames are shown. **C** Quantification of class abundance. **D** Same but for a mouse. The left / right eyes are classified separately. **E** Quantification and sample frames of segmentation. No data is shown if no pixels are detected (mouse is grooming and a limb is obstructing the eyes). The corresponding movies are shown in supplementary movie M5.

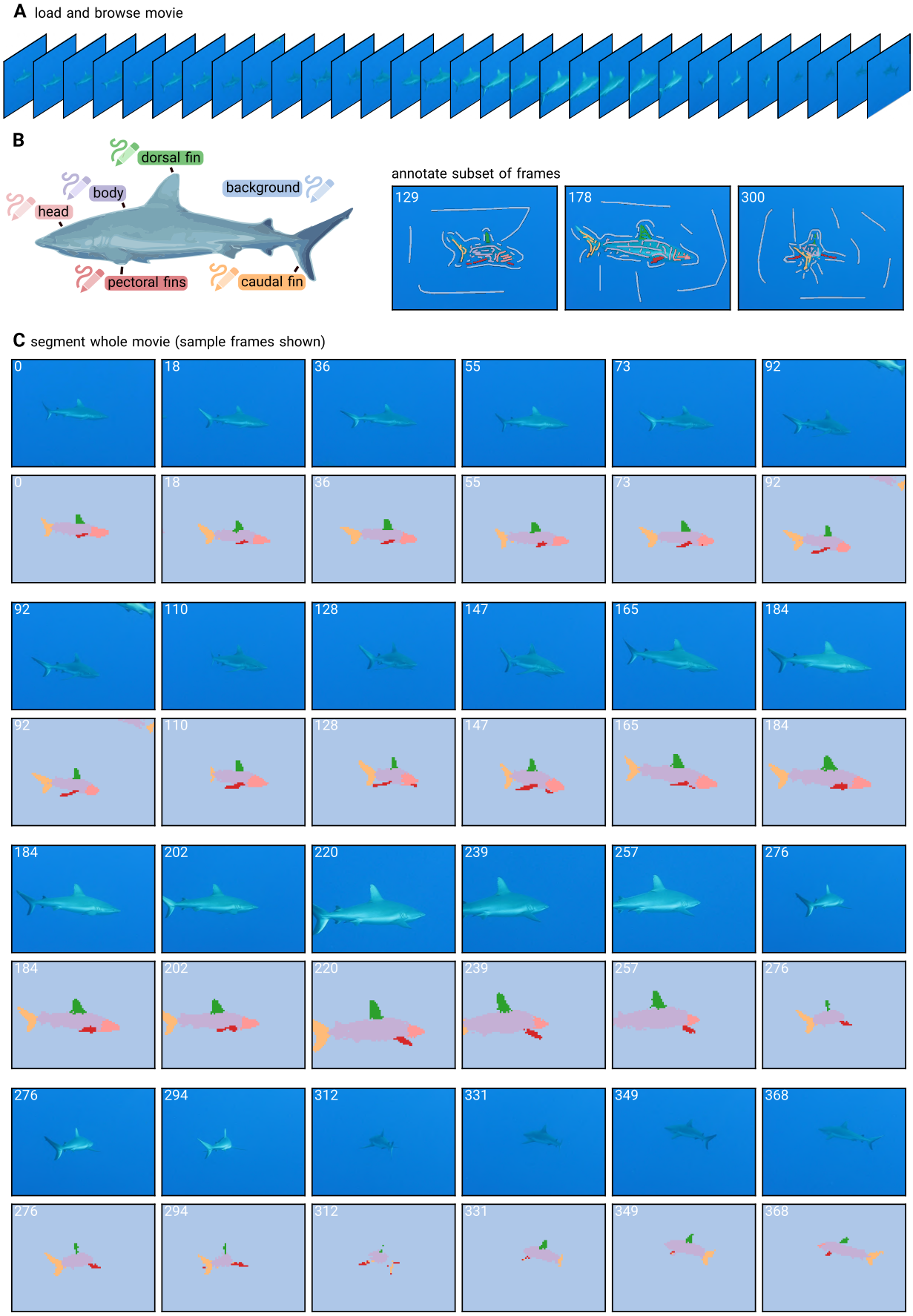

Fig. S7. Detecting shark body parts in video. A The movie shows a shark swimming from multiple angles. Camera is hand-held. **B** Convpaint with DINOv2 as feature extractor is trained on scribbles on three different frames. Head, body, dorsal, caudal, and pectoral fins as well as the background are annotated. **C** The trained model is used to predict the rest of the movie (see supplementary movie M4).

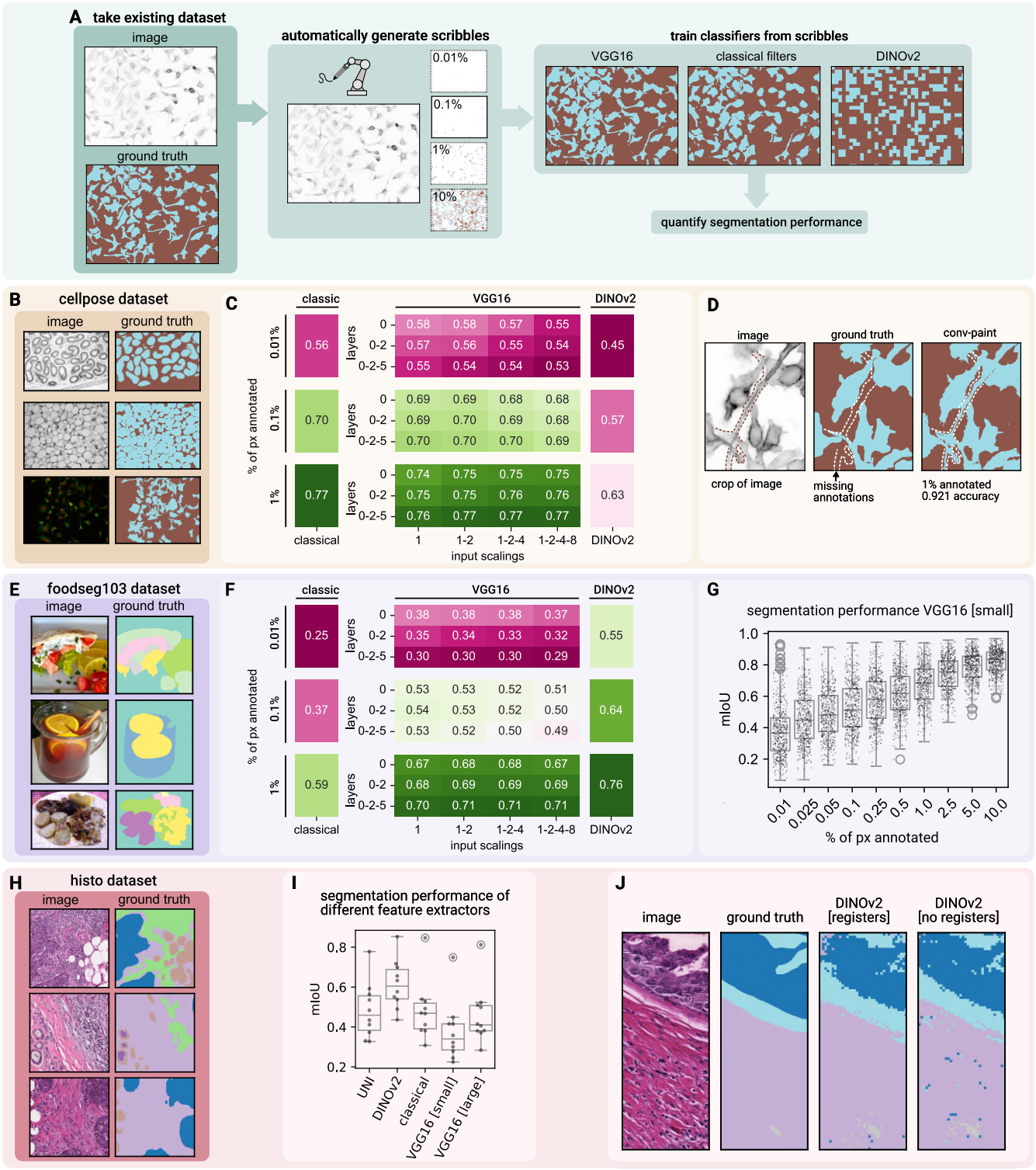

Fig. S8. Quantification of segmentation accuracy and model comparison. **A** Ground truth datasets are used to automatically generate scribbles to train Convpaint, which is then evaluated against the ground truth to quantify the performance of different feature extractors. Various feature extractors are tested, including VGG16 layers with different input scalings, classical filterbanks with all napari-ilastik filter/sigma combinations, and DINOv2 ViT-S/14. **B** Testing on the cellpose dataset (three sample images and masks shown). See figure S9 for segmentation results. **C** Mean mIoU scores for different annotation levels in the cellpose dataset. Pre-trained filters outperform classical filterbanks only in low annotation regimes. DINOv2 underperforms VGG16 and classical filterbanks due to large patch size limitations in capturing small cellular details. **D** Model performance can be limited by the quality of ground-truth annotations. A cell protrusion, missing in the ground truth, is correctly segmented by the model. **E** Testing on the foodseg103 dataset (three samples shown). See fig. S10 for segmentation results. **F** Mean mIoU scores for different annotation levels on the foodseg103 dataset. Pre-trained filters outperform classical filterbanks across all annotation regimes, with DINOv2 performing best. The model is less affected by patch size because of larger regions of interest. **G** For all models, diminishing returns on segmentation performance can be observed with increasing annotation levels, here shown for VGG16 (layer 0 and input scalings 1,2) on the foodseg103 dataset. **H** Testing on a histology slide dataset. See fig. S11 for segmentation results. **I** Mean mIoU scores for different annotation levels in the histology slide dataset. DINOv2 outperforms all other models, including the histology-specific model UNI. **J** DINOv2 with registers produces predictions with less patch noise (not quantified).

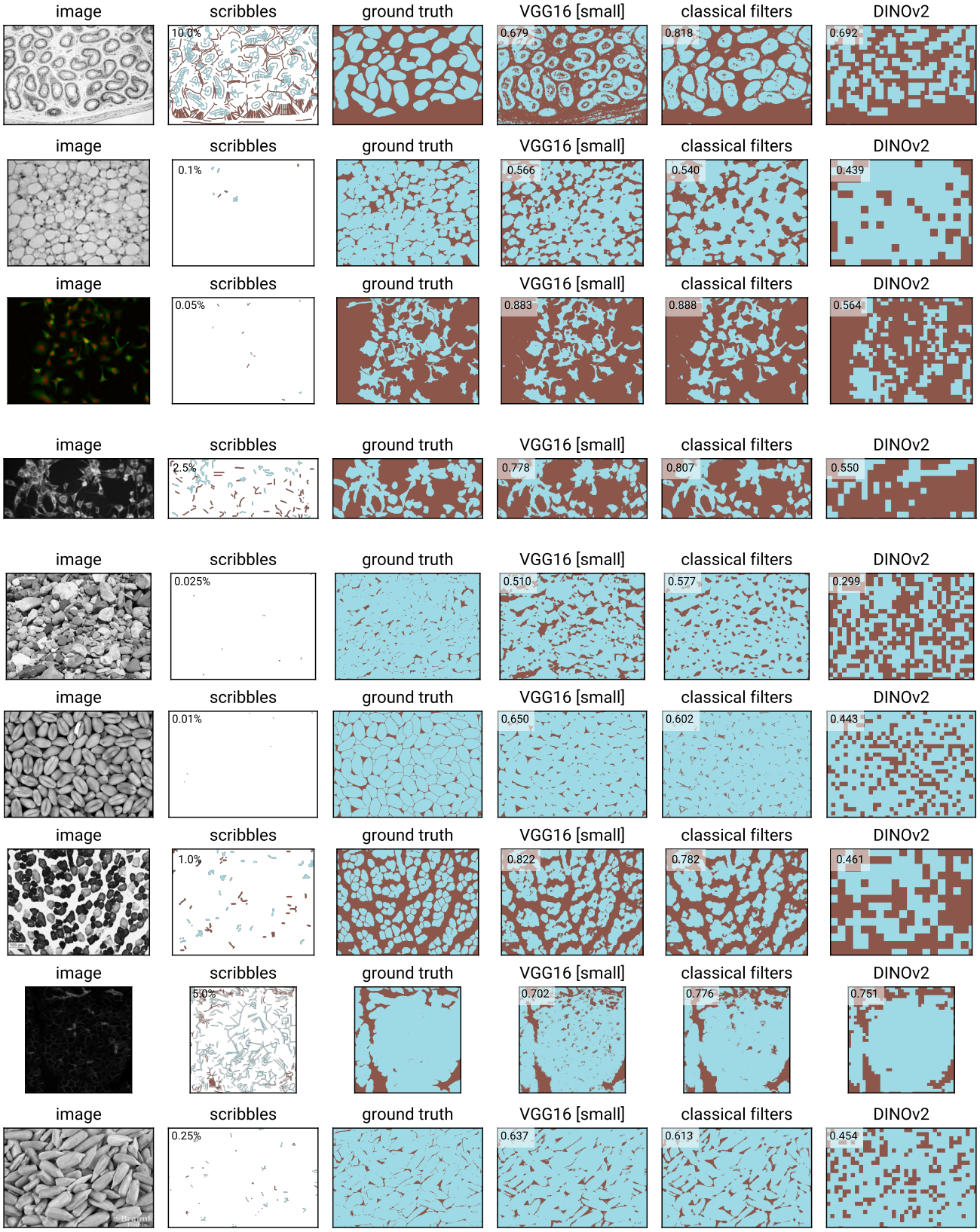

Fig. S9. Feature extractor performance on the Cellpose dataset. Randomly selected images. For plotting, the scribbles were dilated for better visibility. The number in the upper left corner of the prediction images shows the mIoU score.

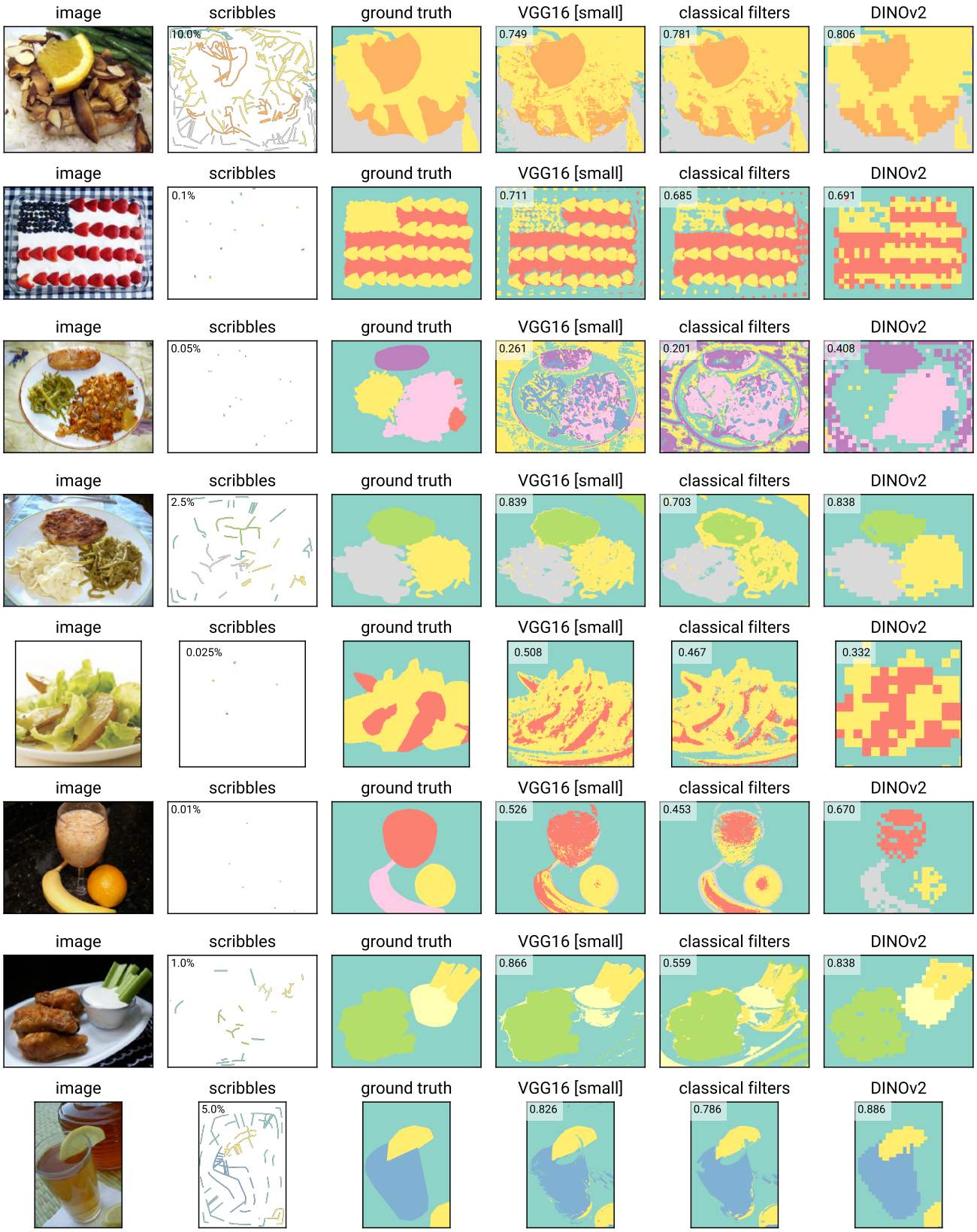

Fig. S10. Feature extractor performance on the foodseg103 dataset. Randomly selected images. For plotting, the scribbles were dilated for better visibility. The number in the upper left corner of the prediction images shows the mIoU score.

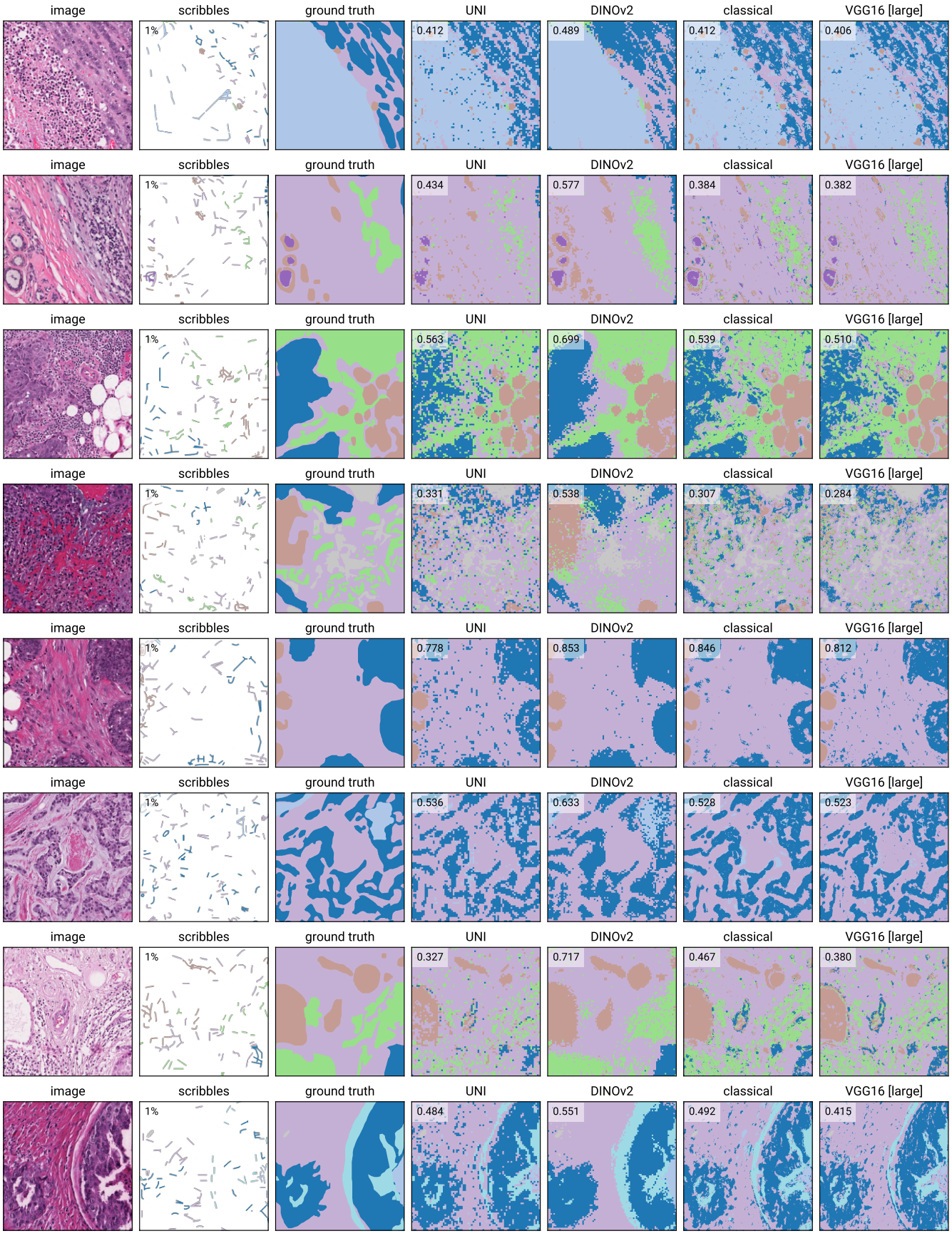

Fig. S11. Feature extractor performance on histology slides. Scribbles were automatically generated from expert annotated ground truth. For plotting, the scribbles were dilated for better visibility. Eight out of ten images used for evaluation in fig. S8l are shown. The number in the upper left corner of the prediction images shows the mIoU score. DINOv2 outperforms all other models in this dataset.

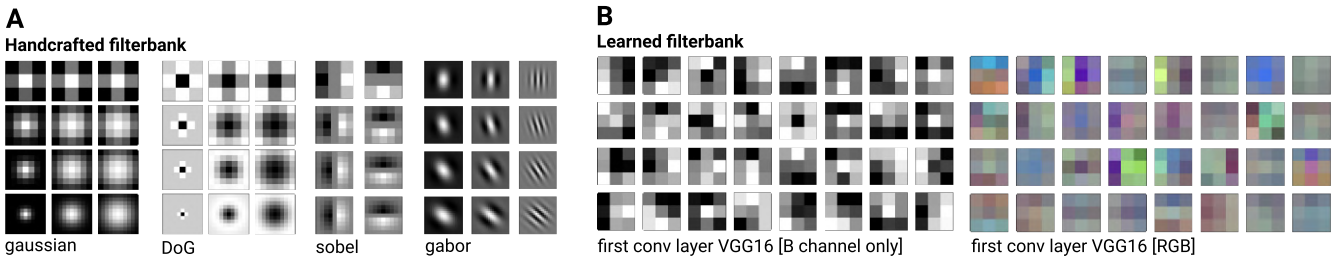

Fig. S12. Visual comparison of handcrafted vs. learned filters A Filters used classically in handcrafted filterbanks. Here we show examples of filter kernels with different parameters for Gaussians, Difference of Gaussians (DoG), Sobel, and Gabor. While these handcrafted filterbanks are more interpretable, the patterns they extract often overlap, leading to redundancy among the filters. **B** Filters extracted from the first convolution layer of a CNN (VGG16) network. The filters have a 3x3x3 shape, which makes them intrinsically capable of extracting correlations between color channels in RGB images. Although VGG16 filters are less interpretable, they are computationally optimized to extract orthogonal image features that are useful for image classification.

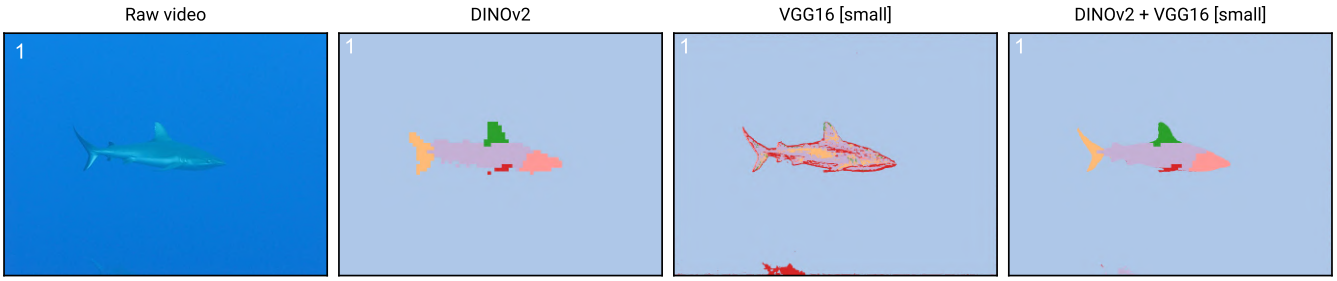

Fig. S13. Combining VGG16 and DINOv2 features for enhanced spatial precision at mask boundaries while maintaining semantic information. While DINOv2 features excell at capturing abstract semantic information at the patch level, VGG16 features are good at capturing local spatial information at the pixel level. By concatenating the features of both models, we can leverage the strengths of both models.