bioRxiv preprint doi: https://doi.org/10.1101/2020.03.20.000133; this version posted March 20, 2020. The copyright holder for this preprint (which was not certified by peer review) is the author/funder, who has granted bioRxiv a license to display the preprint in perpetuity. It is made available under aCC-BY 4.0 International license.

ZeroCostDL4Mic: an open platform to simplify access and use of Deep-Learning in Microscopy

Lucas von Chamier[1 *], Johanna Jukkala[2-3 *], Christoph Spahn[4], Martina Lerche[2], Sara Hernández-Pérez[2,5], Pieta K. Mattila[2,5], Eleni Karinou[6], Seamus Holden[6], Ahmet Can Solak[7], Alexander Krull[8-10], Tim-Oliver Buchholz[8-10], Florian Jug[8-10], Loïc A Royer[7], Mike Heilemann[4], Romain F. Laine[1,11] ✉, Guillaume Jacquemet[2-3] ✉, and Ricardo Henriques[1,11] ✉

[1]MRC-Laboratory for Molecular Cell Biology. University College London, London, UK
[2]Turku Bioscience Centre, University of Turku and Åbo Akademi University, Turku, Finland
[3]Faculty of Science and Engineering, Cell Biology, Åbo Akademi University, Turku, Finland
[4]Institute of Physical and Theoretical Chemistry, Goethe-University Frankfurt, Frankfurt, Germany
[5]Institute of Biomedicine, and MediCity Research Laboratories, University of Turku, Finland
[6]Centre for Bacterial Cell Biology, Biosciences Institute, Faculty of Medical Sciences, Newcastle University, UK
[7]Chan Zuckerberg Biohub, San Francisco, CA, USA
[8]Center for Systems Biology Dresden (CSBD), Dresden, Germany
[9]Max Planck Institute for Molecular Cell Biology and Genetics, Dresden, Germany
[10]Max Planck Institute for Physics of Complex Systems, Dresden, Germany
[11]The Francis Crick Institute, London, UK
[*]Equal contributing authors

Deep Learning (DL) methods are increasingly recognised as powerful analytical tools for microscopy. Their potential to outperform conventional image processing pipelines is now well established (1, 2). Despite the enthusiasm and innovations fuelled by DL technology, the need to access powerful and compatible resources, install multiple computational tools and modify code instructions to train neural networks all lead to an accessibility barrier that novice users often find difficult to cross. Here, we present ZeroCostDL4Mic, an entry-level teaching and deployment DL platform which considerably simplifies access and use of DL for microscopy. It is based on Google Colab which provides the free, cloud-based computational resources needed. ZeroCostDL4Mic allows researchers with little or no coding expertise to quickly test, train and use popular DL networks. In parallel, it guides researchers to acquire more knowledge, to experiment with optimising DL parameters and network architectures. We also highlight the limitations and requirements to use Google Colab. Altogether, ZeroCostDL4Mic accelerates the uptake of DL for new users and promotes their capacity to use increasingly complex DL networks.

Microscopy | Deep Learning | Machine Learning | Google Colab | Segmentation | Denoising| Image Restoration | Artificial Labelling

Correspondence: ***r.laine@ucl.ac.uk, guillaume.jacquemet@abo.fi and r.henriques@ucl.ac.uk***

ZeroCostDL4Mic is a collection of self-explanatory Jupyter Notebooks for Google Colab, featuring an easy-to-use graphical user interface (Fig. S1). It complements current community efforts to simplify access to DL in microscopy, e.g. ImJoy (3) or integration projects of DL into Fiji/ImageJ (4–7). It differs from these solutions by providing a single simple interface to install, use and learn about popular DL networks, while exploiting free high-performance computational resources provided by Google Colab (Fig. 1a). Using ZeroCostDL4Mic does not require prior knowledge of coding since researchers can, in a few mouse clicks and with the help of an easily accessible workflow, install all needed software dependencies, upload their imaging data and run networks for training and inference (Video S1). The underlying code is hidden by default, but remains accessible allowing advanced users to explore and edit the programmatic structure of the notebooks. No extra resources beyond a web browser and a Google Drive account are needed, all computations are directly performed on the cloud using Google's computational capacity.

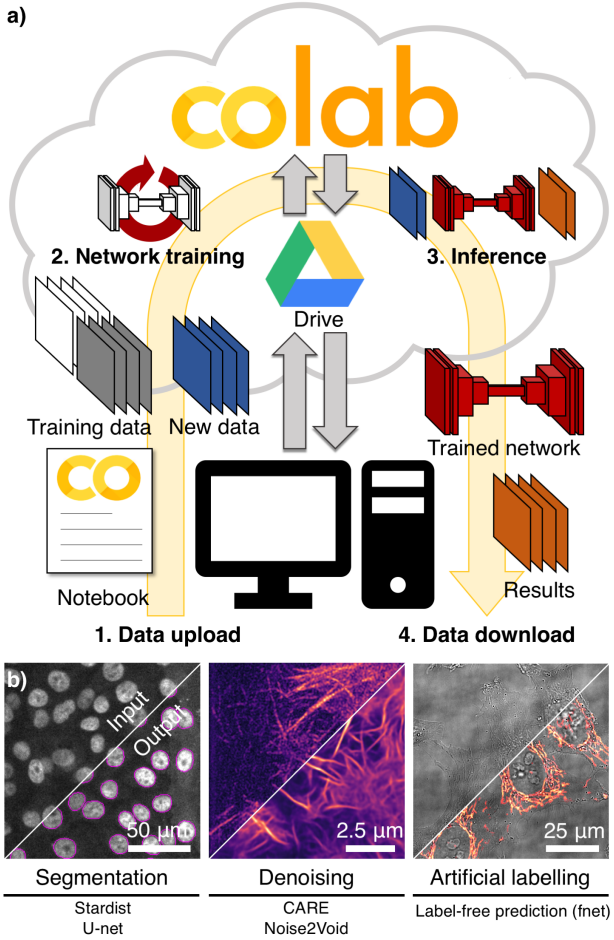

Fig. 1. Overview of ZeroCostDL4Mic a) Workflow of ZeroCostDL4Mic with data transfer via Google Drive and training and inference on Google Colab. b) Overview of the networks currently implemented in ZeroCostDL4Mic and their functions.

bioRxiv preprint doi: https://doi.org/10.1101/2020.03.20.000133; this version posted March 20, 2020. The copyright holder for this preprint (which was not certified by peer review) is the author/funder, who has granted bioRxiv a license to display the preprint in perpetuity. It is made available under aCC-BY 4.0 International license.

We created a common and straightforward interface for popular microscopy DL networks: U-net (8, 9), Stardist (5, 6), CARE (6), Noise2Void (10) and Label-free prediction (fnet) (11). Together, they exploit DL to provide the capacity to run tasks of image segmentation, denoising, restoration and artificial labelling (e.g. prediction of pseudo-fluorescence images from bright-field images) (Fig. 1b and S2, Video S2). Google Colab presents constraints in its free resources, notably RAM capacity and runtime duration (discussed in detail in Supp. Note 1). These, however, do not limit the applicability of ZeroCostDL4Mic for the example datasets we provide and their corresponding networks (Supp. Table S1). These only take a few minutes to hours to train and produce the output shown in Fig. S2 (time taken in Supp. Table S2). In addition, users can easily test the performance of the various networks on their own data.

By bringing previously published methods into a streamlined format that allows easy access and customised use of DL in microscopy, we believe this resource will be an enabling step towards widening the use of DL approaches beyond the community of computer scientists (further discussed in Supp. Note 2). In addition, we envision that the template presented here could be used by DL developers to showcase their own network architectures (Supp. Note 3). This would ensure the rapid dissemination of novel technologies and provide consistent user experience for reproducible and comparative studies of DL approaches.

Availability. ZeroCostDL4Mic is available for download from https://github.com/HenriquesLab/ZeroCostDL4Mic. This resource is fully open-source, providing users with tutorials, Jupyter Notebooks for Google Colab and many real-life example datasets available for training and testing.

ACKNOWLEDGEMENTS

First and foremost, we want to thank Dr Martin Weigert from the Swiss Federal Institute of Technology (EPFL) and Dr Uwe Schmidt from the Max Planck Institute of Molecular Cell Biology and Genetics (MPI-CBG), who pioneered a considerable portion of the technology this work is based on and whose ethos in making Deep Learning more accessible for microscopy helped inspire this work. Dr Schmidt has kindly given key feedback during the preparation of the manuscript. All the network architectures and tasks presented here originate from already published work, having been edited and prepared for Google Colab and to simplify their uptake by novice users. **When using the ZeroCostDL4Mic platform, please cite the original publications associated to each network**.

This work was funded by grants from the UK Medical Research Council (MR/K015826/1) (R.H.) and the Wellcome Trust (203276/Z/16/Z) (R.H.). R.F.L. would like to acknowledge the support of the MRC Skills development fellowship. This work was also supported by grants awarded by the Academy of Finland (to G.J. and P.K.M.), the Sigrid Juselius Foundation (to G.J. and P.K.M.), the University of Turku foundation (to SHP), Åbo Akademi University Research Foundation (CoE CellMech; G.J.) and by Drug Discovery and Diagnostics strategic funding to Åbo Akademi University (G.J.). We thank Dr Aki Stubb for providing us with the raw data used to showcase Noise2Void 2D. The Cell Imaging and Cytometry Core facility (Turku Bioscience, University of Turku, Åbo Akademi University and Biocenter Finland) and Turku Bioimaging are acknowledged for services, instrumentation, and expertise. M.L is supported by Victoriastiftelsen (FI). A.K, T.-O.B and F.J. funded by German Research Foundation (DFG) under the code JU3110/1-1 and German Federal Ministry of Research and Education under the code 01IS18026C, ScaDS2. M.H. and C.S. acknowledge funding by the German Science Foundation (grant nr. SFB1177), C.S. additionally acknowledges support by the European Molecular Biology Organization (short term fellowship 8589). We additionally thank the Chan Zuckerberg Biohub and its donors for funding the work of Loic A. Royer and Ahmet Can Solak. **Note:** CoLaboratory™ and Google Drive™ are trademarks of Google LLC - ©2018 Google LLC All rights reserved.

EXTENDED AUTHOR INFORMATION

- Lucas von Chamier: 0000-0002-9243-912X; ChamierLucas
- Johanna Jukkala: Liehuletti
- Christoph Spahn: 0000-0001-9886-2263; miCHRIScopy
- Martina Lerche: 0000-0002-8203-1120; martina_lerche
- Sara Hernández-Pérez: 0000-0002-0227-1045; sara_mattilalab
- Pieta K. Mattila: 0000-0003-2805-0686; mattilalab
- Eleni Karinou: 0000-0002-1099-4964; Kar__El
- Seamus Holden: 0000-0002-7169-907X; seamus_holden
- Ahmet Can Solak: 0000-0002-1381-8309; _ahmetcansolak
- Alexander Krull: sagzehn
- Tim-Oliver Buchholz: 0000-0001-6953-8915; tibuch_
- Florian Jug: 0000-0002-8499-5812; florianjug
- Loïc A Royer: 0000-0002-9991-9724; loicaroyer
- Mike Heilemann: 0000-0002-9821-3578
- Romain F. Laine: 0000-0002-2151-4487; LaineBioImaging
- Guillaume Jacquemet: 0000-0002-9286-920X; guijacquemet
- Ricardo Henriques: 0000-0002-2043-5234; HenriquesLab

AUTHOR CONTRIBUTIONS

R.L., G.J. and R.H. initiated the research. L.v.C., J.J., T.-O.B., R.L. and G.J. wrote source code based on the work of A.K., T.-O.B., F.J. and L.A.R. among others; L.v.C., J.J., C.S., R.L. and G.J. performed the image acquisition of the test and example data; L.v.C., J.J., C.S., M.L., S.H.-P., P.K.M., E.K., S.H., A. C. S., A.K., T.-O. B., M.H., R.L. and G.J. tested the platform; L.v.C., J.J., C.S., R.L., G.J. and R.H. wrote the manuscript with input from all co-authors.

COMPETING FINANCIAL INTERESTS

We provide a platform based on Google Drive and Google Colab to streamline the implementation of common Deep Learning analysis of microscopy data. Despite heavily relying on Google products, we have no commercial or financial interest in promoting and using these products. In particular, we did not receive any compensation in any form from Google for this work. The authors declare no competing financial interests.

Bibliography

1. Chinmay Belthangady and Loic A Royer. Applications, promises, and pitfalls of deep learning for fluorescence image reconstruction. *Nature methods*, pages 1–11, 2019.
2. Lucas von Chamier, Romain F Laine, and Ricardo Henriques. Artificial intelligence for microscopy: what you should know. *Biochemical Society Transactions*, 47(4):1029–1040, 2019.
3. Wei Ouyang, Florian Mueller, Martin Hjelmare, Emma Lundberg, and Christophe Zimmer. Imjoy: an open-source computational platform for the deep learning era. *Nature methods*, 16(12):1199–1200, 2019.
4. Estibaliz Gómez-de Mariscal, Carlos García-López-de Haro, Laurène Donati, Michael Unser, Arrate Muñoz-Barrutia, and Daniel Sage. Deepimagej: A user-friendly plugin to run deep learning models in imagej. *bioRxiv*, page 799270, 2019.
5. Uwe Schmidt, Martin Weigert, Coleman Broaddus, and Gene Myers. Cell detection with star-convex polygons. In *International Conference on Medical Image Computing and Computer-Assisted Intervention*, pages 265–273. Springer, 2018.
6. Martin Weigert, Uwe Schmidt, Robert Haase, Ko Sugawara, and Gene Myers. Star-convex polyhedra for 3d object detection and segmentation in microscopy. In *The IEEE Winter Conference on Applications of Computer Vision*, pages 3666–3673, 2020.
7. Martin Weigert, Uwe Schmidt, Tobias Boothe, Andreas Müller, Alexandr Dibrov, Akanksha Jain, Benjamin Wilhelm, Deborah Schmidt, Coleman Broaddus, Siân Culley, et al. Content-aware image restoration: pushing the limits of fluorescence microscopy. *Nature methods*, 15(12):1090–1097, 2018.
8. Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. In *International Conference on Medical image computing and computer-assisted intervention*, pages 234–241. Springer, 2015.
9. Thorsten Falk, Dominic Mai, Robert Bensch, Özgün Çiçek, Ahmed Abdulkadir, Yassine Marrakchi, Anton Böhm, Jan Deubner, Zoe Jäckel, Katharina Seiwald, et al. U-net: deep learning for cell counting, detection, and morphometry. *Nature methods*, 16(1):67–70, 2019.
10. Alexander Krull, Tim-Oliver Buchholz, and Florian Jug. Noise2void-learning denoising from single noisy images. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2129–2137, 2019.
11. Chawin Ounkomol, Sharmishtaa Seshamani, Mary M Maleckar, Forrest Collman, and Gregory R Johnson. Label-free prediction of three-dimensional fluorescence images from transmitted-light microscopy. *Nature methods*, 15(11):917–920, 2018.

bioRxiv preprint doi: https://doi.org/10.1101/2020.03.20.000133; this version posted March 20, 2020. The copyright holder for this preprint (which was not certified by peer review) is the author/funder, who has granted bioRxiv a license to display the preprint in perpetuity. It is made available under aCC-BY 4.0 International license.

Methods

Cell Culture. U-251 glioma cells were grown in DMEM/F-12 (Dulbecco's Modified Eagle's Medium/Nutrient Mixture F-12; Life Technologies, 10565-018) supplemented with 10 % fetal bovine serum (FCS) (Biowest, S1860). U-251 glioma cells expressing endogenously tagged paxillin-GFP were generated using CRISPR / Cas9 and were described previously (1). MCF10 DCIS.COM (DCIS.COM) lifeact-RFP cells were cultured in a 1:1 mix of DMEM (Sigma-Aldrich) and F12 (Sigma-Aldrich) supplemented with 5% horse serum (16050-122; GIBCO BRL), 20 ng/ml human EGF (E9644; Sigma-Aldrich), 0.5 mg/ml hydrocortisone (H0888-1G; Sigma-Aldrich), 100 ng/ml cholera toxin (C8052-1MG; Sigma-Aldrich), 10 µg/ml insulin (I9278-5ML; Sigma-Aldrich), and 1% (vol/vol) penicillin/streptomycin (P0781-100ML; Sigma-Aldrich). DCIS.COM lifeact-RFP cells were generated using lentiviruses, produced using pCDH-LifeAct-mRFP, psPAX2, and pMD2.G constructs - see (2) for more details. HeLa ATCC cells were seeded on fibronectin-coated 8-well chamberslides (Sarstedt, Germany, 1.5 x 104 cells/well). Cells were grown for 16h at 37°C and 5% CO2 in Dulbecco's modified Eagle's medium containing 4.5 g/l glucose, 10% FBS and 1% L-alanyl-L-glutamine (Thermo Fisher, GlutaMAX). To fix the HeLa cells, we employed a protocol shown to preserve the cytoskeleton and organelles (adapted from (3)). In brief, the culture medium was directly removed with PHEM buffer containing 3% methanol-free formaldehyde (Thermo Fisher, USA) and 0.2% EM-grade glutaraldehyde (Electron Microscopy Sciences, USA) and incubated the samples for 1 h at room temperature. Cells were washed thrice with PBS, quenched with 0.2% sodium borohydrate in PBS for 7 min and washed again thrice with PBS. A2780 cells were cultured in RPMI 1640 supplemented with 10% FCS. The cells were grown at 37°C in a 5% CO2 incubator.

U-net training dataset. U-net (4) is a deep learning architecture originally developed for segmentation of EM images although it has since been adapted for various other tasks (e.g. CARE, Label-free prediction). Here, we use U-net for segmentation of EM images on the dataset from the ISBI challenge 2012 (5). However, datasets for segmentation tasks can also be created manually. This requires target images which have been segmented by an expert using drawing tools, e.g. in ImageJ/Fiji (6), to draw outlines around the structures of interest. For training in the notebook the source (raw EM image) and target (8-bit mask obtained from expert drawing) images were placed in separate folders, with each source image having a corresponding target image with the same name.

Stardist training dataset. Stardist (7) is a deep learning method that was designed specifically to segment cell nuclei in microscopy images. Training a Stardist network requires matching images of nuclei and of corresponding masks. To generate the nuclei images that we provide as a training dataset, DCIS.COM lifeact-RFP cells were incubated for 2h with 0.5 µM SiR-DNA (SiR-Hoechst, Tetu-bio, Cat Number: SC007) before being imaged live using a spinning-disk confocal microscope. The spinning-disk confocal microscope used was a Marianas spinning disk imaging system with a Yokogawa CSU-W1 scanning unit on an inverted Zeiss Axio Observer Z1 microscope (Intelligent Imaging Innovations, Inc.) equipped with a 20x (NA 0.8) air, Plan Apochromat objective (Zeiss). The corresponding mask images were generated manually in Fiji (6). Briefly, the outlines of each nucleus were drawn using the freehands selection tool and added to the ROI manager. Once all outlines are stored in the ROI manager, the LOCI plugin was used to create a ROI map. The ROI map images were used as the mask images to train Stardist.

Noise2Void training datasets. Noise2Void (8) is a deep learning method that was designed to perform denoising on microscopy images. It takes an unsupervised training approach so it can learn how to denoise a dataset directly from the dataset to denoise. Therefore, no specific training datasets are required, only noisy images and even one noisy image is sufficient to train the network. The 2D dataset provided with our notebooks was generated by plating U-251 glioma cells expressing endogenously tagged paxillin-GFP on fibronectin-coated poly-acrylamide gels (stiffness 9.6 kPa) (1). Cells were then recorded live using a spinning disk confocal microscope equipped with a long working distance 63x (NA 1.15 water, LD C-Apochromat) objective (Zeiss). The 3D dataset provided with our notebooks was generated by recording A2780 ovarian carcinoma cell, transiently expressing Lifeact-RFP (to visualize the actin cytoskeleton), migration on fibroblast-generated cell-derived-matrices, further see (9) for methods. The cell-derived-matrices were labeled using Alexa Fluor 488 recombinant fibronectin and the images acquired using a spinning disk confocal microscope equipped with a 63x oil (NA 1.4 oil, Plan-Apochromat, M27 with DIC III Prism) objective (Zeiss). For both datasets, the spinning disk confocal microscope used was a Marianas spinning disk imaging system with a Yokogawa CSU-W1 scanning unit on an inverted Zeiss Axio Observer Z1 microscope controlled by SlideBook 6 (Intelligent Imaging Innovations, Inc.). Images were acquired using a Photometrics Evolve, back-illuminated EMCCD camera (512 x 512 pixels).

Content-aware image restoration (CARE) training datasets. CARE (10) is a deep learning method capable of image restoration from corrupted bio-images (whether corrupted by noise, artefacts or low resolution for instance). The network allows image denoising and resolution improvement in 2D and 3D images, using supervised training. The specific function of the network is determined by the images provided in the training dataset. For instance, if noisy images are provided as input and high signal-to-noise ratio images are provided as targets, the network will perform denoising. The dataset provided as an example with our notebooks was generated to denoise live-cell structured illumination microscopy (SIM) imaging data. Briefly, DCIS.COM lifeact-RFP cells were plated on high tolerance glass-bottom dishes (MatTek

bioRxiv preprint doi: https://doi.org/10.1101/2020.03.20.000133; this version posted March 20, 2020. The copyright holder for this preprint (which was not certified by peer review) is the author/funder, who has granted bioRxiv a license to display the preprint in perpetuity. It is made available under aCC-BY 4.0 International license.

Corporation, coverslip 1.7) pre-coated first with Poly-L lysine (10 mg/ml, 1 h at 37°C) and were allowed to reach confluence. Cells were then fixed and permeabilized simultaneously using a solution of 4% (wt/vol) paraformaldehyde and 0.25% (vol/vol) Triton X-100 for 10 min. Cells were then washed with PBS, quenched using a solution of 1 M glycine for 30 min, and incubated with phalloidin-488 (1/200 in PBS; Cat number: A12379; Thermo Fisher scientific) at 4°C until imaging (overnight). Just before imaging using SIM, samples were washed three times in PBS and mounted in vectashield (Vectorlabs). The SIM used was DeltaVision OMX v4 (GE Healthcare Life Sciences) fitted with a 60x Plan-Apochromat objective lens, 1.42 NA (immersion oil RI of 1.516) used in SIM illumination mode (five phases 3 three rotations). Emitted light was collected on a front illuminated pco.edge sCMOS (pixel size 6.5 mm, readout speed 95 MHz; PCO AG) controlled by SoftWorx. In the provided dataset, the high signal-to-noise ratio images were acquired from the phalloidin-488 staining using acquisition parameters optimal to obtain the best SIM images possible (in this case, 50 ms exposure time, 10% laser power). In contrast the low signal-to-noise ratio images were acquired from the LifeAct-RFP channel using acquisition parameters more suitable for live-cell imaging (in this case, 100 ms exposure time, 1% laser power). The dataset provided with the 2D CARE notebooks are maximum intensity projections of the collected data.

Label-free prediction (fnet) training dataset. Fnet (11) is a deep learning method which was developed as a tool for label-predictions from unannotated brightfield and EM images. Training fnet in the provided Colab notebook requires 3D stacks in two channels, e.g. fluorescence and transmitted light. Before acquisition fixed HeLa ATCC cells were permeabilised and blocked using 0.25% TX-100 (Sigma Aldrich, Germany) and 3% IgG-free BSA (Carl Roth, Germany) in PBS for 1,5 h. Cells were labelled for TOM20 using 5 µg/ml rabbit anti-TOM20 primary antibody (sc-11415, Santa Cruz, USA) and 10 µg/ml donkey anti-rabbit-secondary antibody (Alexa Fluor 594 conjugated, A32754, Thermo Fisher, USA) in PBS containing 0.1% TX-100 and 1% BSA for 1,5 h each. Samples were rinsed twice and washed 3x with PBS (5 min) after each incubation step. Image stacks were acquired on a Leica SP8 confocal microscope (Leica Microsystems, Germany) bearing a 63x/1.40NA oil objective (Leica HC PL APO). The pixel size was set to 90 nm in xy-dimensions and 150 nm in z (32 slices) and fluorescence image stacks were recorded using 561 nm laser excitation and collected by a PMT. The corresponding transmitted light image stack was recorded in parallel using a transmitted light PMT. Of the acquired 3D stacks 23 were used for training and 2 for testing (see Supp. Table S1). The raw data was converted into .tif file format and split into stacks of the respective channels (fluorescence and transmitted light). To prepare a training set, stacks were split into individual folders by channel. The signal files (transmitted light) and their respective targets (fluorescence) must be in the same order in their respective folders. The code requires matching files to be in a corresponding position in their respective folders to create matched training pairs. It is therefore advisable to number source-target pairs or to give the files the same names. This was done with an ImageJ/Fiji macro (6).

Bibliography

1. Aki Stubb, Romain F Laine, Mitro Miihkinen, Hellyeh Hamidi, Camilo Guzmán, Ricardo Henriques, Guillaume Jacquemet, and Johanna Ivaska. Fluctuation-based super-resolution traction force microscopy. *Nano Letters*, 2019.
2. Guillaume Jacquemet, Ilkka Paatero, Alexandre F Carisey, Artur Padzik, Jordan S Orange, Hellyeh Hamidi, and Johanna Ivaska. Filoquant reveals increased filopodia density during breast cancer progression. *Journal of Cell Biology*, 216(10):3387–3403, 2017.
3. Wesley R Legant, Lin Shao, Jonathan B Grimm, Timothy A Brown, Daniel E Milkie, Brian B Avants, Luke D Lavis, and Eric Betzig. High-density three-dimensional localization microscopy across large volumes. *Nature methods*, 13(4):359–365, 2016.
4. Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. In *International Conference on Medical image computing and computer-assisted intervention*, pages 234–241. Springer, 2015.
5. Ignacio Arganda-Carreras, Srinivas C Turaga, Daniel R Berger, Dan Cireşan, Alessandro Giusti, Luca M Gambardella, Jürgen Schmidhuber, Dmitry Laptev, Sarvesh Dwivedi, Joachim M Buhmann, et al. Crowdsourcing the creation of image segmentation algorithms for connectomics. *Frontiers in neuroanatomy*, 9:142, 2015.
6. Johannes Schindelin, Ignacio Arganda-Carreras, Erwin Frise, Verena Kaynig, Mark Longair, Tobias Pietzsch, Stephan Preibisch, Curtis Rueden, Stephan Saalfeld, Benjamin Schmid, et al. Fiji: an open-source platform for biological-image analysis. *Nature methods*, 9(7): 676–682, 2012.
7. Martin Weigert, Uwe Schmidt, Robert Haase, Ko Sugawara, and Gene Myers. Star-convex polyhedra for 3d object detection and segmentation in microscopy. In *The IEEE Winter Conference on Applications of Computer Vision*, pages 3666–3673, 2020.
8. Alexander Krull, Tim-Oliver Buchholz, and Florian Jug. Noise2void-learning denoising from single noisy images. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2129–2137, 2019.
9. Riina Kaukonen, Guillaume Jacquemet, Hellyeh Hamidi, and Johanna Ivaska. Cell-derived matrices for studying cell proliferation and directional migration in a complex 3d microenvironment. *nature protocols*, 12(11):2376, 2017.
10. Martin Weigert, Uwe Schmidt, Tobias Boothe, Andreas Müller, Alexandr Dibrov, Akanksha Jain, Benjamin Wilhelm, Deborah Schmidt, Coleman Broaddus, Siân Culley, et al. Content-aware image restoration: pushing the limits of fluorescence microscopy. *Nature methods*, 15(12):1090–1097, 2018.
11. Chawin Ounkomol, Sharmishtaa Seshamani, Mary M Maleckar, Forrest Collman, and Gregory R Johnson. Label-free prediction of three-dimensional fluorescence images from transmitted-light microscopy. *Nature methods*, 15(11):917–920, 2018.

bioRxiv preprint doi: https://doi.org/10.1101/2020.03.20.000133; this version posted March 20, 2020. The copyright holder for this preprint (which was not certified by peer review) is the author/funder, who has granted bioRxiv a license to display the preprint in perpetuity. It is made available under aCC-BY 4.0 International license.

Supplementary Note 1: Limitations of Google Colab

The Google Colab platform offers a free and straightforward access to a Graphical Processing Unit (GPU) and Tensor Processing Unit (TPU), which significantly lowers the entry barrier for new users of Deep Learning methods. However, this access comes with drawbacks, especially in terms of reliability of the access to GPUs and limitations of storage and RAM. In our hands, we consistently found that GPU acceleration provided faster computation that TPU for the networks and datasets presented. Therefore we focused our attention on GPU accessibility and performance below. Please note, however, that these limitations can be largely alleviated by two main approaches:
- **None of these limitations applies when connecting to a local runtime, i.e. running the notebooks on the user's workstation**. This option allows the user to access their local files directly from the Google Colab notebooks and does not have any of the limitations outlined below. However, if the user does not have GPU access locally, running networks will likely be significantly slower than using a GPU runtime on Colab. Making use of the local runtime requires some additional steps outlined in the Colab manual shown in https://research.google.com/colaboratory/faq.html.
- The limitations can be largely alleviated by upgrading to Google Colab Pro for a small financial investment.

1.1. Limited free Google Drive storage. If using a free Google Drive account to perform training, the user will have access to 15 GB of data storage that can be accessed by any Google Colab notebooks. All training and test datasets, plus output of the predictions, need to fit within this 15 GB limit. We have shown, however, that this is sufficient to train all of the networks with the datasets that we provide (Supp. Table S1). Additional storage space can be purchased from Google.

1.2. GPU and TPU RAM limits. A 12.72 GB RAM limit currently exists for the free GPU or TPU provided by Google Colab. The amount of RAM required to execute the code is determined by the size of the training data, the number and sizes of patches/batches. Exceeding this RAM limit can cause the notebook to crash or show an error when creating the datasets within the notebook or when loading data into a network for training. For the datasets we have tested, it was always possible to train the networks with the currently available RAM. However, large datasets (especially the 3D ones) may reach or exceed the RAM capacity when using a large number of patches/batches.

1.3. Time-outs.

12-hour time-out. The time taken to train a network sufficiently is primarily determined by the size of the training dataset, the number of steps/epochs/patches as well as the efficiency of the underlying code. Google Colab currently offers 12 hours of GPU/TPU access after which local variables and data loaded into the network are deleted, a limit primarily enforced to prevent cryptocurrency-mining. For users, this means that if training has not completed by the 12h limit, all progress may be lost if no intermediate steps were saved. This constraint can be problematic when networks need to be trained over many epochs to reach high performance, often necessary for large datasets. We have found that Label-free prediction (fnet) typically requires long training times (Supp. Table S2), compared to the other networks tested here. Users can circumvent this issue by saving network weights in a Google Drive folder regularly. In the Label-free prediction (fnet) notebook, checkpoints are automatically saved which allows the user to continue training later from such a checkpoint after a potential time-out.

Log-out if idle. Google Colab may disconnect significantly earlier than 12 hours if it detects an interruption on user interaction or network training. Usually, this time-out happens after 30 to 90 minutes of idleness in our hands, i.e. code not running or lack of user interaction with the Google Colab interface. When the log-out occurs, local variables, installed packages and data stored outside any mounted drive are deleted. Hence, if the log-out occurs before training a model, cells setting up parameters for training such as paths or hyperparameters may have to be reset before training. With all of our notebooks, the models are automatically saved in Google Drive upon training completion, meaning that long training sessions do not have to be attended.

1.4. Unreliable GPU access. Google Colab does not guarantee access to a GPU, as the number of users of the service may be larger than the number of available devices. It is not clear how access to a GPU is regulated, but it may be determined by traffic to the Google Colab servers. If no GPUs are available, Colab will offer to run the notebook using either on a TPU, on CPU (without acceleration) or as a local run-time (i.e. on the machine of the user). However, these alternatives can be significantly slower than Google Colab GPU access. However, if no GPU is currently available, GPU access usually becomes available again after a few hours or a day.

1.5. GPU type. Google Colab uses different GPUs which currently include Nvidia K80, T4, P4 and P100 (as of March 2020). The user cannot decide which GPU will be available when using the notebook. According to the Google Colab FAQ, this is due to limitations in the provision of a free service to users which makes certain types of GPU unavailable at the time a notebook is used. In practice, this should not affect the performance of the networks in the notebooks. However, it may affect the speed at which networks can be trained and used. Therefore, users might encounter variability in training times as a consequence. To find out which type of GPU Google Colab is using, the user can play the first cell in each notebook which will give information on GPU access and type.

bioRxiv preprint doi: https://doi.org/10.1101/2020.03.20.000133; this version posted March 20, 2020. The copyright holder for this preprint (which was not certified by peer review) is the author/funder, who has granted bioRxiv a license to display the preprint in perpetuity. It is made available under aCC-BY 4.0 International license.

1.6. How to mitigate the limitations of Google Colab. Several steps can be taken by users of ZeroCostDL4Mic to mitigate Google Colab limitations. Regarding the 12h training limitation, we encourage users to change the number of epochs and other training parameters so that the training can be completed in less than 12h. This is readily possible with all of the networks we provide. In the case of Label-free prediction (fnet), only the number of epochs can be changed to shorten the training time so, in order to accommodate large number of epochs, we provide the option to continue the training from a saved training checkpoint in case of a time-out. This allows to start from a pre-trained model and accumulate many rounds of epochs.

For the log-out if idle issue, Google Colab cells can be played all at once (or a subset at once). In this case, the activated cells will run one after the other. This can be useful to ensure that the run-time does not disconnect until everything has been completed and the user's data is safely saved in google drive.

ZeroCostDL4Mic is aimed to be an entry point to learn about DL methods where users can easily train networks with their own data. While ZeroCostDL4Mic is completely free to use, many of its limitations (associated with the Google Colab platform) can be mitigated with small financial investments. For instance the 15 GB google drive limitation can be easily improved by purchasing more Google Drive storage from Google. In addition, Google is now rolling out a Google Colab Pro version that offers faster GPUs, longer runtimes and access to more RAM. If these intermediate options are still not sufficient, ZeroCostDL4Mic notebooks can also be easily adapted to run on the user's own computer by connecting Google Colab to a local runtime. This of course requires the users to invest in a powerful workstation.

bioRxiv preprint doi: https://doi.org/10.1101/2020.03.20.000133; this version posted March 20, 2020. The copyright holder for this preprint (which was not certified by peer review) is the author/funder, who has granted bioRxiv a license to display the preprint in perpetuity. It is made available under aCC-BY 4.0 International license.

Supplementary Note 2: When in doubt, always retrain! A supplementary discussion

The primary focus of ZeroCostDL4Mic is to provide a straightforward and free platform to aid novice users using Deep Learning in microscopy. A vital component of this platform is the capacity to simplify network training, which remains a significant existing limitation in the field. Because it can be difficult to train networks, many labs are taking the approach of providing pre-trained network models that can be used to process imaging data. Both DeepImageJ (1) and CSBDeep (2) are excellent examples. This tends to create model "zoos" - databases of pre-trained models loadable into software packages such as DeepImageJ and CSBDeep. However, pre-trained models can be very powerful, they can also be very specific to the microscopes and sample types used in their training and may lead to erroneous/artefactual results when applied to widely different dataset type than that it was training on. As a result, it is not uncommon for researchers to apply inappropriate pre-trained models to their data, which unfortunately lead to inaccurate but often visually pleasing results (3, 4).
Given this issue, it becomes critical for researchers to have the option to train models using their own specific data in order to produce high-fidelity and reliable results. Here, users face several constraints, such as the difficulty to install the right dependencies and the need to have access to powerful computational resources. ZeroCostDL4Mic significantly lowers these barriers by helping researchers access the resources needed to train, in an accessible manner. We believe ZeroCostDL4Mic takes a significant step to increase the uptake of DL tools by microscopy users and highlights the need for all DL users to take the step towards understanding and tackling the limitations of the approach.

Supplementary Note 3: Future perspectives for ZeroCostDL4Mic

Together with the help of the wider research community, we expect to grow the number of networks available in ZeroCostDL4Mic quickly. We are currently working in collaboration with the Royer laboratory to add Noise2Self (5) to this repertoire. We also expect ZeroCostDL4Mic to become a standard which users can use to evolve existing networks, adapting them to data analysis tasks optimised for their specific image processing problems. To this end, we will further keep developing ZeroCostDL4Mic, continuously adapting it to novel Deep Learning paradigms. This focus will incorporate the capacity to maintain compatibility with rapidly evolving Deep Learning libraries, and the ability to export pre-trained models which could be used in either DeepImageJ, CSBDeep or other inference engines.

bioRxiv preprint doi: https://doi.org/10.1101/2020.03.20.000133; this version posted March 20, 2020. The copyright holder for this preprint (which was not certified by peer review) is the author/funder, who has granted bioRxiv a license to display the preprint in perpetuity. It is made available under aCC-BY 4.0 International license.

Supplementary Table 1: Training Datasets

Network	Data Type	# of files	Image Sizes	Image Type	Comments
U-net	Training - Images	28	512x512	EM 8-bit TIFF	ISBI or here
U-net	Training - Masks	28	512x512	Binary 8-bit TIFF	ISBI or here
U-net	Test - Images	2	512x512	EM 8-bit TIFF	ISBI or here
U-net	Test - Masks	2	512x512	EM 8-bit TIFF	ISBI or here
Stardist	Training - Images	45	1024x1024	Fluo 16-bit TIF	This paper
Stardist	Training - Masks	45	1024x1024	Object-labelled 8-bit TIF	This paper
Stardist	Test - Images	2	1024x1024	Fluo 16-bit TIF	This paper
Stardist	Test - Masks	2	1024x1024	Object labelled 8-bit TIF	This paper
Stardist	Test - Stacks	4	1024x1024x86	Fluo 16-bit TIF	This paper
N2V (2D)	Training - Images	1	512x512	Fluo 16-bit TIFF	Stubb et al, 2020 (6)
N2V (2D)	Test - Images	22	512x512	Fluo 16-bit TIFF	Stubb et al, 2020 (6)
N2V (3D)	Training - Images	1 (Actin) +1 (Fibronectin)	512x512x13	Fluo 16-bit TIFF	Kaukonen et al, 2017 (7) (Actin and fibronectin datasets)
N2V (3D)	Test - Images	48 (Actin) +48 (Fibronectin)	512x512x13	Fluo 16-bit TIFF	Kaukonen et al, 2017 (7) (Actin and fibronectin datasets)
CARE (2D)	Training – Low SNR images	22	1024x1024	SIM fluo (MIP from 3D stack) 32-bit TIFF	This paper (Filopodia dataset - Maximum projection)
CARE (2D)	Training – High SNR images	22	1024x1024	SIM fluo (MIP from 3D stack) 32-bit TIFF	This paper (Filopodia dataset - Maximum projection)
CARE (2D)	Test – Low SNR images	2	1024x1024	SIM fluo (MIP from 3D stack) 32-bit TIFF	This paper (Filopodia dataset - Maximum projection)
CARE (2D)	Test – High SNR images	2	1024x1024	SIM fluo (MIP from 3D stack) 32-bit TIFF	This paper (Filopodia dataset - Maximum projection)
CARE (3D)	Training – Low SNR images	22	1024x1024x33	SIM fluo 32-bit TIF	This paper (Filopodia dataset 3D - stack)
CARE (3D)	Training – High SNR images	22	1024x1024x33	SIM fluo 32-bit TIF	This paper (Filopodia dataset 3D - stack)
CARE (3D)	Test – Low SNR images	2	1024x1024x33	SIM fluo 32-bit TIF	This paper (Filopodia dataset 3D - stack)
CARE (3D)	Test – High SNR images	2	1024x1024x33	SIM fluo 32-bit TIF	This paper (Filopodia dataset 3D - stack)
Label-free prediction (fnet)	Training - Brightfield	23	1024x1024x32	Bright-field confocal 8-bit TIF	This paper
Label-free prediction (fnet)	Training – Fluo (mitochondrial marker)	23	1024x1024x32	Fluo confocal 8-bit TIF	This paper
Label-free prediction (fnet)	Test - Brightfield	2	1024x1024x32	Bright-field confocal 8-bit TIF	This paper
Label-free prediction (fnet)	Test – Fluo (mitochondrial marker)	2	1024x1024x32	Fluo confocal 8-bit TIF	This paper

Table S1. Overview of the datasets used for training the networks. In all cases of supervised network training (all networks provided here apart from Noise2Void), the test datasets consist of the last two files generated for training. These were set aside for testing and are not part of the training dataset.

Supplementary Table 2: Example network training times with indicated parameters and GPU access

Network Name	Training Epochs	Steps	Batch size	Image Dimensions	# of images	# of Patches, patch size, patch height	GPU type	Time for training (using these settings)
U-net	200	6	4	1024x1024	28	n.a.	Tesla P100-PCIE-16GB	8 min
Stardist	400	12	4	1024x1024	45	n.a.	Tesla P100-PCIE-16GB	14 min
N2V (2D)	100	61	128	512x512	22	392, 64	Tesla T4	17 min
N2V (3D)	100	133	128	512x512x13	48	392, 64, 8	Tesla P100-PCIE-16GB	240 min
CARE (2D)	50	31	64	1024x1024	22	100, 80	Tesla P100-PCIE-16GB	3 min
CARE (3D)	50	62	64	1024x1024x33	22	200, 80, 8	Tesla P4	90 min
Label-free prediction (fnet)	n.a.	50000	4	1024x1024x32	23	n.a.	Tesla P4	8h20min

Table S2. Example network settings and training times for each notebook.

bioRxiv preprint doi: https://doi.org/10.1101/2020.03.20.000133; this version posted March 20, 2020. The copyright holder for this preprint (which was not certified by peer review) is the author/funder, who has granted bioRxiv a license to display the preprint in perpetuity. It is made available under aCC-BY 4.0 International license.

Supplementary Figure 1: Colab graphical user interface (GUI)

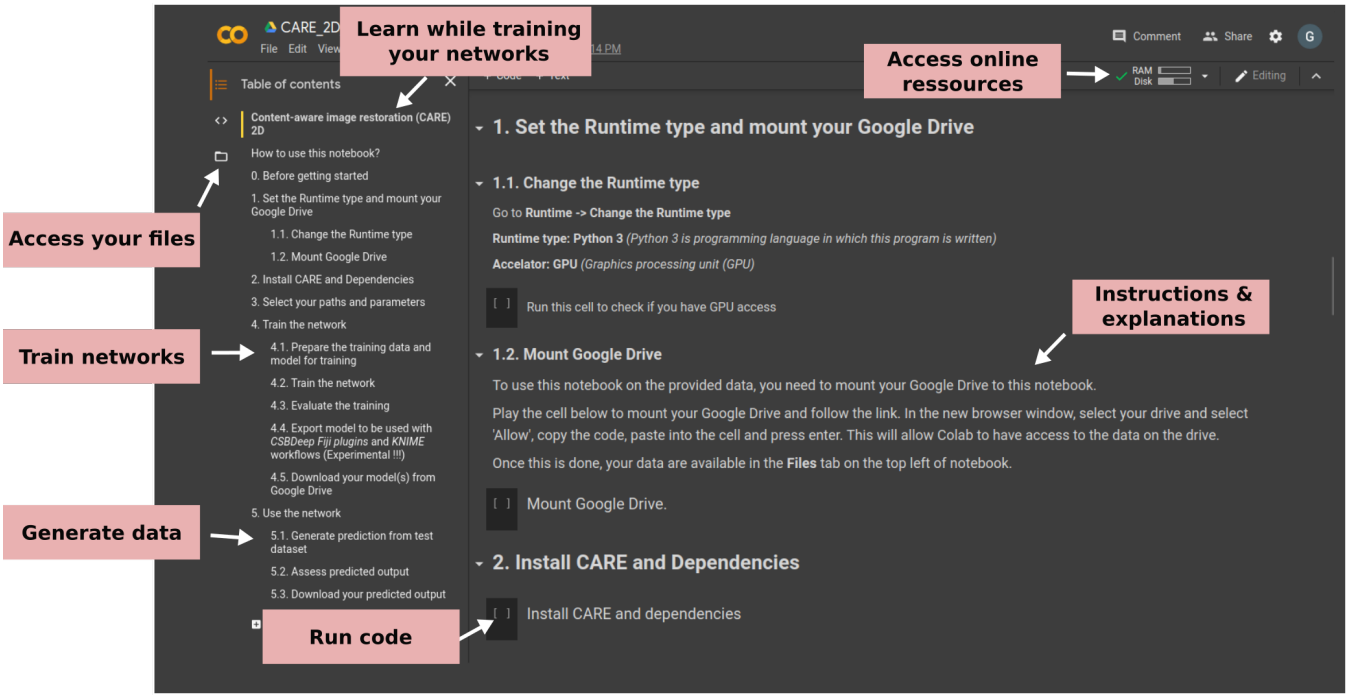

Fig. S1. Graphical user interface (GUI) of the ZeroCostDL4Mic notebooks. The layout of the notebook and quick access to the different sections is available on the left panel. The user has access to the files present on their Google Drive.

Supplementary Figure 2: Network input and output examples

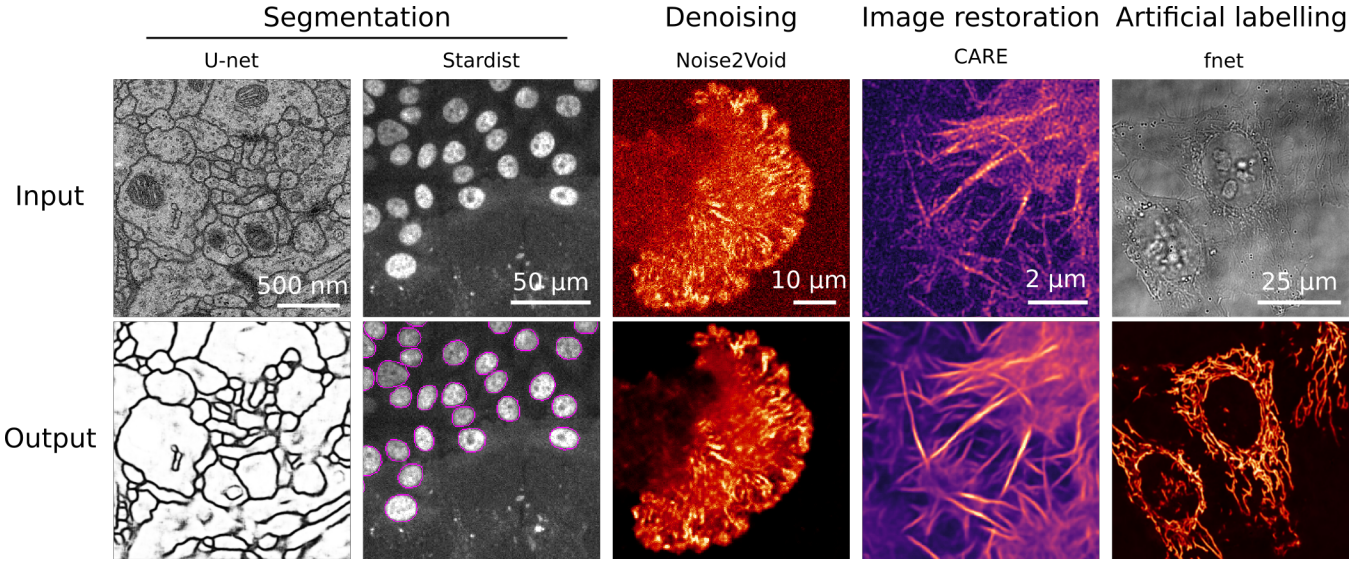

Fig. S2. Overview of the tasks possible to perform with ZeroCostDL4Mic and the corresponding networks. Overview of the networks currently implemented in ZeroCostDL4Mic and their tasks.

bioRxiv preprint doi: https://doi.org/10.1101/2020.03.20.000133; this version posted March 20, 2020. The copyright holder for this preprint (which was not certified by peer review) is the author/funder, who has granted bioRxiv a license to display the preprint in perpetuity. It is made available under aCC-BY 4.0 International license.

Supplementary Videos

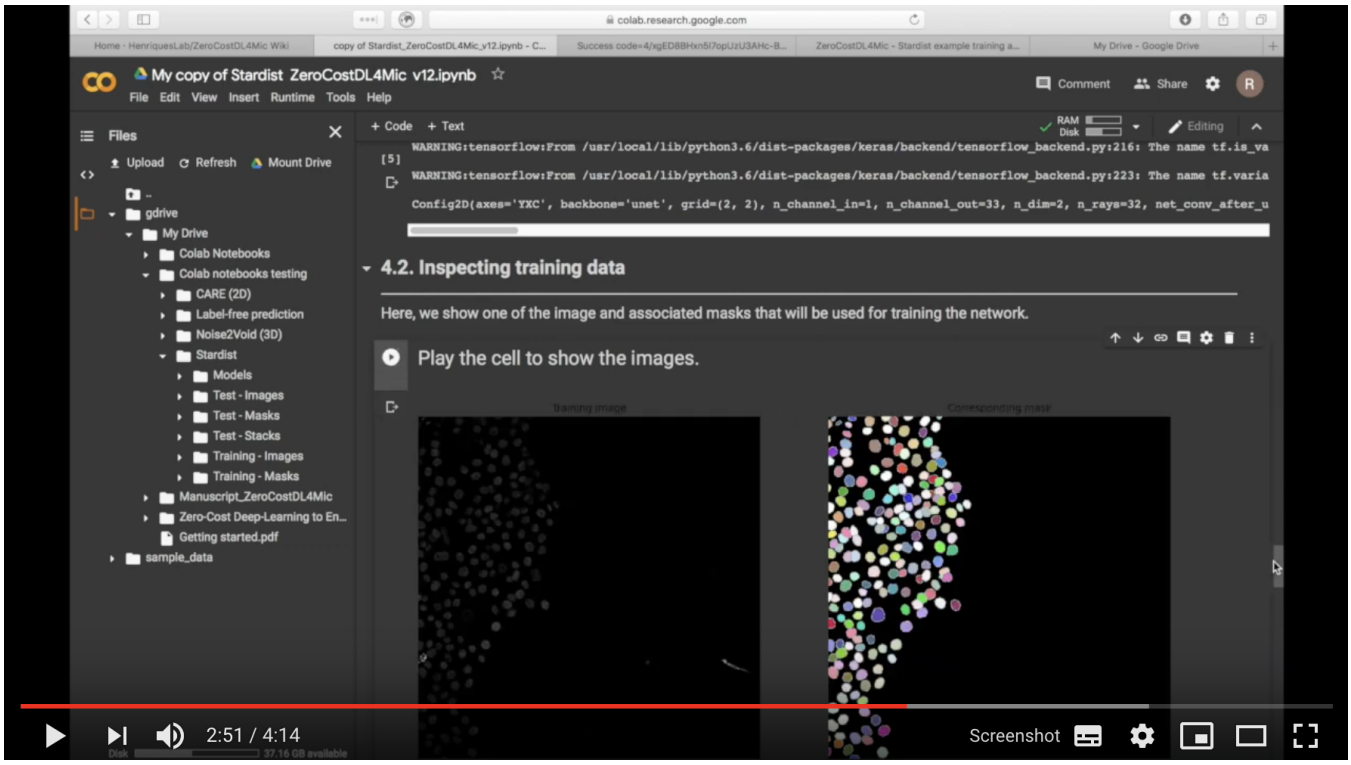

Fig. S3. Supplementary Video 1. Full run through of the workflow to obtain the notebooks and the provided test datasets as well as a common use of the notebook. YouTube link: https://youtu.be/GzD2gamVNHI.

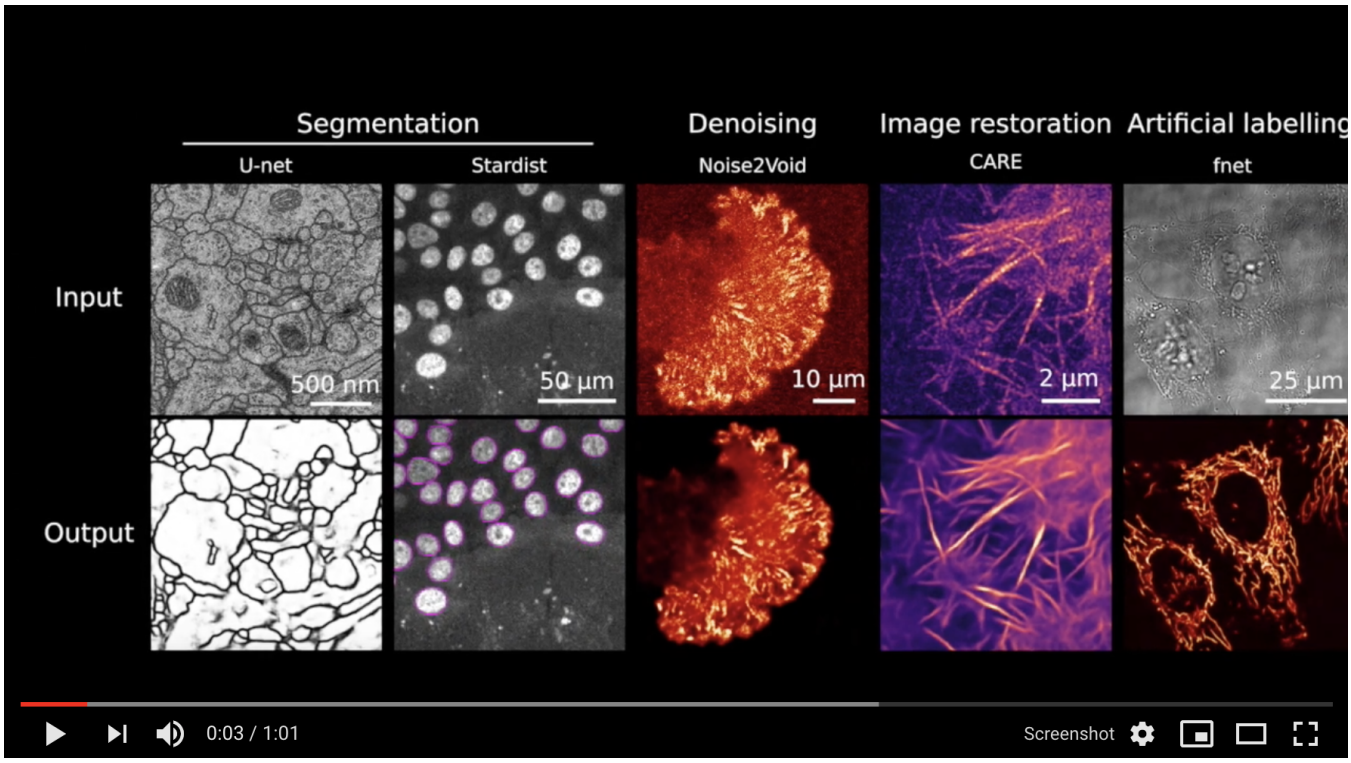

Fig. S4. Supplementary Video 2. Representative results obtained from the provided test dataset. YouTube link: https://youtu.be/hh2I5xJH67k.

bioRxiv preprint doi: https://doi.org/10.1101/2020.03.20.000133; this version posted March 20, 2020. The copyright holder for this preprint (which was not certified by peer review) is the author/funder, who has granted bioRxiv a license to display the preprint in perpetuity. It is made available under aCC-BY 4.0 International license.

Supplementary Bibliography

1. Estibaliz Gómez-de Mariscal, Carlos García-López-de Haro, Laurène Donati, Michael Unser, Arrate Muñoz-Barrutia, and Daniel Sage. Deepimagej: A user-friendly plugin to run deep learning models in imagej. *bioRxiv*, page 799270, 2019.
2. Martin Weigert, Uwe Schmidt, Tobias Boothe, Andreas Müller, Alexandr Dibrov, Akanksha Jain, Benjamin Wilhelm, Deborah Schmidt, Coleman Broaddus, Siân Culley, et al. Content-aware image restoration: pushing the limits of fluorescence microscopy. *Nature methods*, 15(12):1090–1097, 2018.
3. Lucas von Chamier, Romain F Laine, and Ricardo Henriques. Artificial intelligence for microscopy: what you should know. *Biochemical Society Transactions*, 47(4):1029–1040, 2019.
4. Chinmay Belthangady and Loic A Royer. Applications, promises, and pitfalls of deep learning for fluorescence image reconstruction. *Nature methods*, pages 1–11, 2019.
5. Joshua Batson and Loic Royer. Noise2self: Blind denoising by self-supervision. *arXiv preprint arXiv:1901.11365*, 2019.
6. Aki Stubb, Romain F Laine, Mitro Miihkinen, Hellyeh Hamidi, Camilo Guzmán, Ricardo Henriques, Guillaume Jacquemet, and Johanna Ivaska. Fluctuation-based super-resolution traction force microscopy. *Nano Letters*, 2019.
7. Riina Kaukonen, Guillaume Jacquemet, Hellyeh Hamidi, and Johanna Ivaska. Cell-derived matrices for studying cell proliferation and directional migration in a complex 3d microenvironment. *nature protocols*, 12(11):2376, 2017.