Technische Universität München
TUM School of Life Sciences

Deep learning methods for single cell morphometry in high-throughput microscopy

Dominik J. E. Waibel, M.Sc.

Vollständiger Abdruck der von der TUM School of Life Sciences der Technischen Universität München zur Erlangung des akademischen Grades eines

Doktors der Naturwissenschaften (Dr. rer. nat.)

genehmigten Dissertation.

Vorsitz: Prof. Dr. Dmitrij Frischmann

Prüfer*innen der Dissertation:

1. Prof. Dr. Dr. Fabian J. Theis

2. Prof. Dr. Julia Schnabel

Die Dissertation wurde am 5.12.2022 bei der Technischen Universität München eingereicht und durch die TUM School of Life Sciences am 19.06.2023 angenommen.

Abstract

For decades, light microscopy has been an inevitable technology in biological and medical research. Recently, the massive increase of data acquired with high temporal and spatial resolution in high-throughput experiments requires new image analysis tools to fully exploit technological advances in microscopy. Deep learning has become an indispensable tool for data analysis in biomedicine as it can outperform established machine learning and analytical methods and human experts, both in terms of data processing speed and in terms of prediction accuracy and prediction consistency. Furthermore, deep learning methods can detect and extract previously inaccessible information that is not obvious to humans, thus enabling new ways for data analysis. However, there is still a need for deep learning methods that ease access to morphological information of a single cell, while reducing imaging time and obviate tissue fixation and staining, and thereby enable high-throughput single cell microscopy. In this thesis, methods that facilitate and accelerate the analysis of three-dimensional (3D) microscopy images are presented: To ease access to state-of-the-art deep learning methods for a broad range of users, without requiring a profound computational background, InstantDL combined basic image processing tools, such as image segmentation, pixel-wise regression and image classification in one easy-to-apply package. Minimizing microscopic image acquisition times, I developed a deep learning model, called SHAPR, that predicts 3D single cell shapes from two-dimensional (2D) microscopy images and allows researchers to shorten acquisition time by a manifold. Utilizing a novel loss function based on topological features, the prediction performance of my model was further improved. In the application of red blood cell classification, SHAPR significantly improved performance in terms of F1 score. Maximizing the amount of information extractable from 3D bright-field time-lapse movies, while reducing laborious and error prone tissue fixation and staining, I developed the deep learning method Bright2Nuc. This deep learning method predicts nuclear stainings from 3D bright-field microscopy images, enabling the tracking and quantification of morphological properties of single cells in bright-field movies. Applied on human-induced pluripotent 3D cell cultures, Bright2Nuc enables the detection of cell differentiation events by tracking of cell mobility in 3D time-lapse bright-field movies, and Bright2Nuc enables the morphological feature based transcription factor prediction in 3D bright-field images.

The presented methods not only extract previously inaccessible information from 3D microscopic images, but also pave the way towards microscopy-based, non-invasive, high-throughput single cell monitoring and image analysis and are suited to process the increasing amount of microscopic data. These methods hence offer researchers the opportunity to upscale 3D cell phenotyping and expand the impact that researchers can generate from their microscopy image data.

Zusammenfassung

Seit Jahrzehnten ist die Lichtmikroskopie eine unverzichtbare Technologie in der biologischen und medizinischen Forschung. In jüngster Zeit erfordert die massive Zunahme von Daten, die mit hoher zeitlicher und räumlicher Auflösung in Hochdurchsatz-Experimenten erfasst werden, neue Methoden für die Bildanalyse, um die technologischen Fortschritte in der Mikroskopie voll auszuschöpfen. Deep Learning ist zu einem unverzichtbaren Werkzeug für die Datenanalyse in der Mikroskopie geworden, da es analytische Methoden und menschliche Experten sowohl in Bezug auf die Geschwindigkeit der Datenverarbeitung als auch in Bezug auf die Vorhersagegenauigkeit und die Konsistenz der Vorhersagen übertreffen kann. Darüber hinaus können Deep-Learning-Methoden durch automatisches Feature-Learning bisher unzugängliche Informationen erkennen und extrahieren, die für den Menschen nicht offensichtlich sind. Es besteht jedoch immer noch Bedarf an Deep-Learning-Methoden, die den Zugang zu morphologischen Informationen einer einzelnen Zelle erleichtern und gleichzeitig die Bildgebungszeit verkürzen und die Fixierung und Färbung des Gewebes überflüssig machen, so dass die Einzelzellmikroskopie mit hohem Durchsatz möglich wird.

In dieser Arbeit werden Methoden vorgestellt, die die Analyse von dreidimensionalen (3D) Mikroskopiebildern erleichtern und beschleunigen: Um den Zugang zu modernen Deep-Learning-Methoden für ein breites Spektrum von Nutzern zu erleichtern, ohne dass ein tiefgreifendes IT Kentnisse vorauszusetzen, sind in InstantDL grundlegende Bildverarbeitungswerkzeuge wie Bildsegmentierung, pixelweise Regression und Bildklassifizierung in einem einfach anzuwendenden Softwarepaket kombiniert. Um die Aufnahmezeiten für mikroskopische Bilder zu minimieren, habe ich ein Deep-Learning-Modell namens SHAPR entwickelt, das 3D-Formen von einzelnen Zellen aus zweidimensionalen (2D) Mikroskopiebildern vorhersagt und es damit Forschern ermöglicht, die Aufnahmezeit um ein Vielfaches zu verkürzen. Durch die Verwendung einer neuartigen Loss-funktion, die auf topologischen Features basiert, konnte die Performance meines Modells weiter verbessert werden. Bei der Anwendung der Klassifizierung roter Blutkörperchen verbesserte SHAPR die Leistung in Bezug auf den F1-Score erheblich. Um die Menge an Informationen zu maximieren, die aus Zeitrafferfilmen von 3D-Hellfeld-Bildern extrahiert werden können, und gleichzeitig die mühsame und fehleranfällige Fixierung und Färbung von Gewebe zu reduzieren, habe ich die Deep-Learning-Methode Bright2Nuc entwickelt. Diese Deep-Learning-Methode sagt Kernfärbungen aus 3D-Hellfeldmikroskopiebildern voraus und ermöglicht die Verfolgung und Quantifizierung morphologischer Eigenschaften einzelner Zellen in Filmen. Angewandt auf humane induzierte pluripotente 3D-Zellkulturen, ermöglicht Bright2Nuc die Erkennung von Zelldifferenzierungen durch Verfolgung der Mobilität der Zellen in Zeitrafferfilmen von 3D-Hellfeldmikroskopen, und Bright2Nuc ermöglicht die auf morphologischen Merkmalen basierende Vorhersage von Transkriptionsfaktoren in 3D-Hellfeldbildern.

Die vorgestellten Methoden extrahieren nicht nur bisher unzugängliche Informationen aus mikroskopischen 3D-Bildern, sondern ebnen auch den Weg für mikroskopie basierte, nicht-invasive Beobachtung einzelner Zellen und Bildanalyse im Hochdurchsatz und sind geeignet, die wachsende

Menge an mikroskopischen Daten zu verarbeiten. Die in dieser Arbeit entwickelten Methoden bieten den Forschern daher die Möglichkeit, die Phänotypisierung zu verbessern und die Information, welche Forscher aus Mikroskopie-Bildern extrahieren können, zu maximieren.

Acknowledgements

First and foremost, I thank Carsten Marr for offering me the PhD position in his team, for his enduring support, for our great scientific discussions, for the exciting projects and for creating an inspiring research environment. His contributions cannot be overestimated, his ideas, knowledge and encouragement have been outstanding. Moreover, I thank Carsten for the freedom to follow my own ideas and for the positive attitude that helped me during my time in his team.

I thank Fabian Theis for enabling my PhD, for his valuable feedback during the last four years and for creating such a fantastic, open and thriving research environment at the Computational Health Center.

I thank Dmitrij Frischmann, Fabian and Julia Schnabel for examining my PhD thesis. And Carsten, Fabian and Henrik Semb for their feedback on my work during the Thesis Advisory Comitee meetings.

Furthermore, I thank Matthias Meier for our fruitful collaboration as most of my work is based on the microscopy data recorded in his team, for the valuable guidance in our weekly meetings and for his support during manuscript writing and submission.

During the collaboration with Scott Atwell, I learned about stem cell biology and microscopy imaging. I thank Scott for imaging the cell cultures for our projects and for his valuable ideas and discussions.

Moreover, I thank Bastian Rieck for our collaboratively implemented topological loss function, for many insightful and enjoyable meetings and for all that I have learned from him.

I thank Raja Giryes for hosting and supervising me during my stay at the Tel Aviv University and for his valuable contribution to our diffusion model. Both the project and life in Israel were an unforgettable experience, it opened many new perspectives to me.

Furthermore, I thank Henrik, for his valuable ideas and support and for the project we started together. Thanks also for inviting Carsten and me to Copenhagen, it was a fantastic visit.

Especially, I thank Ali Bousheri, Sophia Wagner, Lorenz Lamm, Ario Sadafi, Benedikt Mairhörmann, Rushin Gindra, Valerio Lupperger, Olle Holmberg, Niklas Köhler, Christian Matek, Alejandra Castelblanco, Benjamin Schubert, Salome Kazeminia, Mohammad Mirkazemi and Tingying Peng for the projects we worked on together.

I was lucky to guide a number of great students through their Bachelor's and Master's theses. Therefore, I thank Niklas Kiermeyer, Tuğba Oktar, Carla Glassl, Ahmad Bin Quasim and Aleksandra Kornivetc for the trust in my supervision. I learned a lot and grew during these projects. I thank Ali, Valerio and Olle for sharing the supervision of some of these students. Also, I thank Ali and Ernst Röell for continuing the projects I began but could not finish.

I warmly thank Johanna Winter, Melanie Schulz, Sophia, Lorenz, Moritz Thomas, Benedikt, Rushin, Valentin Koch and Ali for the great scientific discussions and for contributing their ideas to my projects. And I thank them for the valuable feedback on my manuscripts and this thesis.

A special thanks goes to Lea Schuh, Olle, Sophie Tritschler and Valerio for their warm welcome and practical guidance at the start of my PhD.

Equally important, I thank the members of the Marr, Peng and Theis labs and all other members of the Computational Health Center for the great working atmosphere, for the continuous commitment, for the great discussions, for the lunch on Thursdays, for the positive spirit and for all our social events.

A special thanks to both the administration at the Computational Health Center, namely Marianne Antunes, Elisabeth Noheimer, Sabine Kunz and Anna Sacher.

I warmly thank my friends and family for their enduring support, not only during this thesis.

Thank you very much.

You all made this journey a very enjoyable and memorable experience and I am looking forward to keeping in touch with you for years to come.

List of contributed publications

This thesis is based on the following core publications as main author:

Core publications as main author

I) Waibel, D. J. E., Boushehri S. S., and Marr, C. (2021). InstantDL - An easy-to-use deep learning pipeline for image segmentation and classification. *BMC Bioinformatics*, *22*(103), doi: https://doi.org/10.1186/s12859-021-04037-3
(See also publication [1] in the bibliography and in A.1.)

II) Waibel, D. J. E., Kiermeyer, N., Atwell, S., Sadafi, A., Meier M., Marr, C. (2022). SHAPR predicts 3D cell shapes from 2D microscopic images. *iScience*, *25*(11), doi:
doi: https://doi.org/10.1016/j.isci.2022.105298
(See also publication [2] in the bibliography and in A.2.)

III) Waibel, D. J. E., Atwell, S., Meier M., Marr, C., Rieck, B. (2022). Capturing Shape Information with Multi-Scale Topological Loss Terms for 3D Reconstruction. *Medical Image Computing and Computer Assisted Intervention – MICCAI 2022*, pp. 150–159,
doi: https://doi.org/10.1007/978-3-031-16440-8_15
(See also publication [3] in the bibliography and in A.3.)

Further manuscript as main author currently in publication

IV) Atwell, S.*, Waibel, D. J. E.*, Boushehri S. S., Marr, C., Meier M. (* contributed equally) (2022). Label-free imaging of 3D pluripotent stem cell differentiation dynamics on chip, *Accepted at Cell Reports Methods*
(See also manuscript [4] in the bibliography and in A.4.)

Further publications not included in this thesis

V) Waibel, D. J. E., Tiemann, U., Lupperger, V., Semb, H., Marr, C., (2019). In-silico staining from bright-field and fluorescent images using deep learning. *International Conference on Artificial Neural Networks (ICANN)*, Lecture Notes in Computer Science, vol 11729. Springer, Cham.
doi: https://doi.org/10.1007/978-3-030-30508-6_15
(See also publication [5] in the bibliography.)

VI) Boushehri, S. S., Qasim, A., Waibel, D. J. E., Schmich, F., Marr, C., (2022). Systematic comparison of incomplete-supervision approaches for biomedical imaging classification. *International Conference on Artificial Neural Networks (ICANN)*, Lecture Notes in Computer Science, vol 13529. Springer, Cham.
doi: https://doi.org/10.1007/978-3-031-15919-0_30
(See also manuscript [6] in the bibliography.)

VII) Wagner, S. J., Matek, C., Boushehri, S. S., Boxberg, M., Lamm, L., Sadafi, A., Waibel, D. J. E., Marr, C., Peng, T., (2022). Make Deep Learning Algorithms in Computational Pathology More Reproducible and Reusable. *Nature Medicine* 28, 1744–1746 (2022),
doi: https://doi.org/10.1038/s41591-022-01905-0
(See also publication [7] in the bibliography.)

VIII) Wagner, S. J., Matek, C., Boushehri, S. S., Boxberg, M., Lamm, L., Sadafi, A., Waibel, D. J. E., Marr, C., Peng, T., (2022). Built to last? Reproducibility and Reusability of Deep Learning Algorithms in Computational Pathology. *medRxiv*,
doi: https://doi.org/10.1101/2022.05.15.22275108
(See also manuscript [8] in the bibliography.)

IX) Mairhörmann, B., Castelblanco, A., Häfner, F., Pfahler, V., Haist, L., Waibel, D. J. E., Flemmer, A., Ehrhardt, H., Stoecklein, S., Dietrich, O., Foerster, K., Hilgendorff, A., Schubert, B., (2021). Deep Learning-Based Magnetic Resonance Imaging Lung Segmentation and Volumetric Marker Extraction in Preterm Infants. *medRxiv*, *Submitted*,
doi: https://doi.org/10.1101/2021.08.06.21261648
(See also manuscript [9] in the bibliography.)

X) Waibel, D. J. E., Röell, E., Rieck, B., Giryes, R., Marr, C., (2022). A Diffusion Model Predicts 3D Shapes from 2D Microscopy Images. *Accepted at IEEE International Symposium on Biomedical Imaging (ISBI)*,
doi: https://doi.org/10.48550/arXiv.2208.14125
(See also manuscript [10] in the bibliography.)

XI) Boushehri, S. S., Kornivetc, A., Waibel, D. J. E., Kazeminia, S., Schmich, F., Marr, C., (2023) PXPermute: Unveiling staining importance in multichannel fluorescence microscopy. *arXiv*,
doi: https://doi.org/10.1007/978-3-031-15919-0_30
(See also manuscript [11] in the bibliography.)

I, Dominik Waibel, am the main author of the manuscripts [1–4]. A summary of these articles can be found in section 1.9 and in full length in the sections 2.1, 2.2, 2.3, and 2.4, respectively. A detailed descriptions of my specific contributions to these manuscripts can be found in section 1.9.

Contents

1. Introduction

Observing and uncovering the morphology of single cells and their dynamic behavior is a central goal in biological research [12, 13]. Since populations of cells are most often heterogeneous in function and fate, a quantitative description at the single cell level allows for insights into the inner workings of cellular dynamics beyond the population level. By measuring phenotypic, spatial and temporal behavior of single cells, researchers can investigate molecular processes, cellular states, differentiation decisions or disease progression [13]. Biomedical research combines the fields of biology and medicine with the aim to understand biological processes to improve medical care.

1.1. Microscopic imaging for single cell analysis

Insights into phenotypic relationships of single cells can be provided by microscopic imaging [13–15]. In an ideal scenario, single cells should be observed *in-vivo* for an extended period of time in an non-invasive manner and with high resolution to gain insights into phenotypic, spatial and temporal behavior of single cells. This requires high-quality imaging and statistical methods for the quantitative analysis of single cells and single cell populations. For decades, microscopy has been used for studying single cells. Since the origin of pathology in 19th century, microscopy has made significant improvements in terms of resolution and throughput [16]. While most of these advances originate from improved hardware, in recent years software that can be used to post-process and analyze the recorded images is now the driver of innovation in microscopic imaging.

Light microscopy has become a standard technique to uncover single cell properties in tissues in the past decades. It allows for insight into spatial properties and dynamics with sub-micrometer resolution and provides spatial and temporal information at a single cell level. The simplest, cheapest and most popular light microscopy technique is bright-field microscopy, as it requires no staining of the sample. With bright-field microscopy, the sample is illuminated with white light, while the contrast in the recorded image is caused by differences in attenuation of the light by the sample. This label-free microscopy is most commonly used for live cell imaging, for example to record time-lapse movies, as it does not require fixation and staining of the sample or reporter cell lines. However, due to the transparency of cells, the detection and quantification of single cells from bright-field images is difficult.

Fluorescence microscopy, the development of fluorescent stains and protein tagging allow researchers to image cellular properties and thereby reveal specific tissue or cell structures as well as morphology. For analyzing cellular processes, fluorescence microscopy has an irreplaceable role as it enables studies of functional processes and morphological aspects of cells. As an example, fluorescent microscopy plays a vital role in drug discovery, where microscopic imaging is used at several stages from preclinical research to clinical trials. Fluorescent microscopy enables scientists to investigate phenotypes, organs, tissues, cells and subcellular components. Enabled by digital image analysis, the biological mechanisms can be revealed [17].

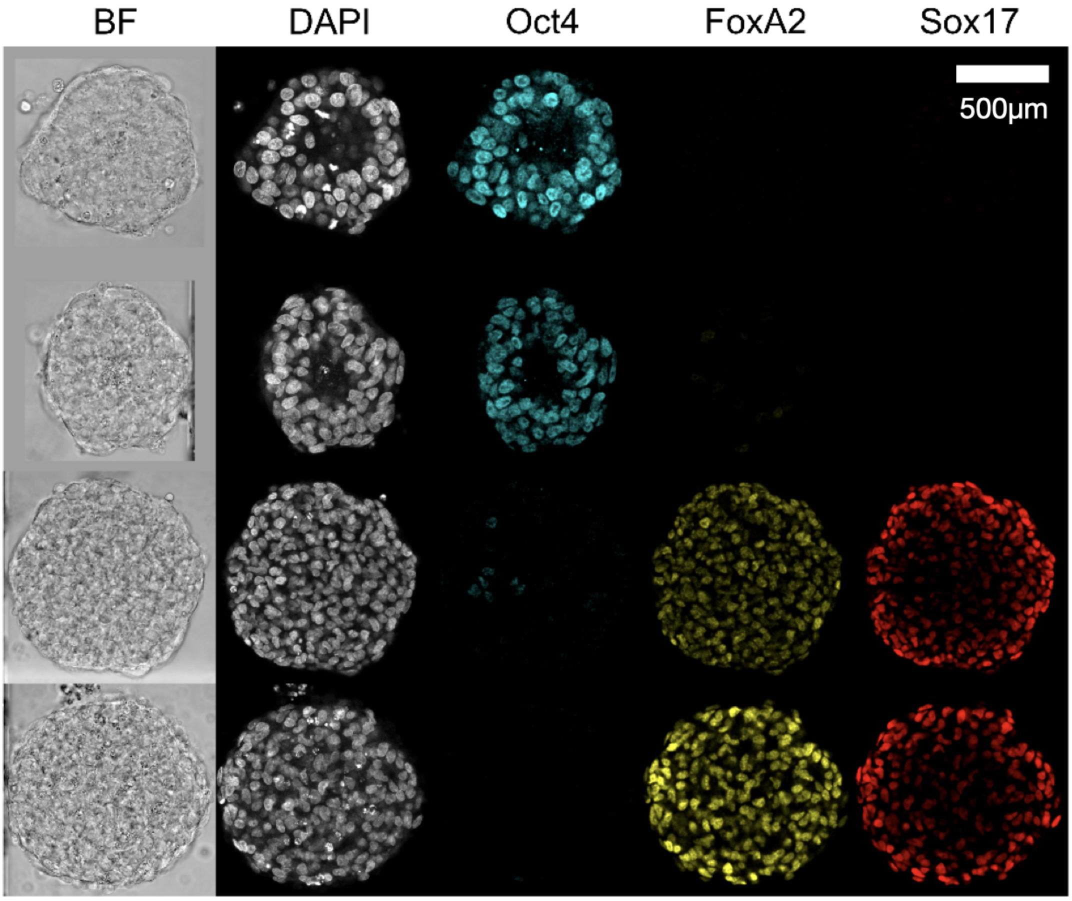


Figure 1.1: Four cell clusters imaged with a confocal microscope. The first column displays the cell cluster imaged in bright-field (BF) and the second the same cell clusters imaged with a fluorescent DAPI stain that binds to the DNA of cells, thus indicating the location of nuclei. Specific stains enable the investigation of cell states, Oct-4 is a fluorescent stain that can be used to mark undifferentiated cells, while FoxA2 and Sox17 indicates differentiated cells.

Tissue samples can have intrinsically fluorescent dyes, such as the green fluorescent protein, or be labeled with a fluorescent marker, which binds to specific sites in the cells, such as the DNA [18]. The contrast in the recorded image is created by the fluorescent dye that emits photons when illuminated with light of the required wavelength, called the excitation light, which is often created by a laser [18]. This allows for quantification of spatial distribution of fluorescent particles that are tagged to cellular regions of interest. Using fluorescent dyes, the traditional resolution limit of optical microscopes, the Abbe limit, can be overcome [19]. However, fluorescence labeling brings a number of disadvantages, such as difficult signal reproducibility and photo-bleaching that describes the effect when a fluorescent dye irreversibly loses its ability to fluoresce due to exposure excitation light [18]. Exposure to the excitation light can impair sample physiology and even lead to sample death by damaging macromolecules. This photo-toxicity frequently occurs during fluorescent microscopy [18, 20]. Biological structures can not be imaged easily *in-vivo*, as it requires reporter cell lines, which are costly to create as stable cell lines that have been labeled with reporter genes have to be breed [13, 14]. To overcome the need to stain biological samples in order to reveal their structures, machine learning researchers are developing methods for single cell microscopy and extend the impact of statistical microscopic image analysis [12].

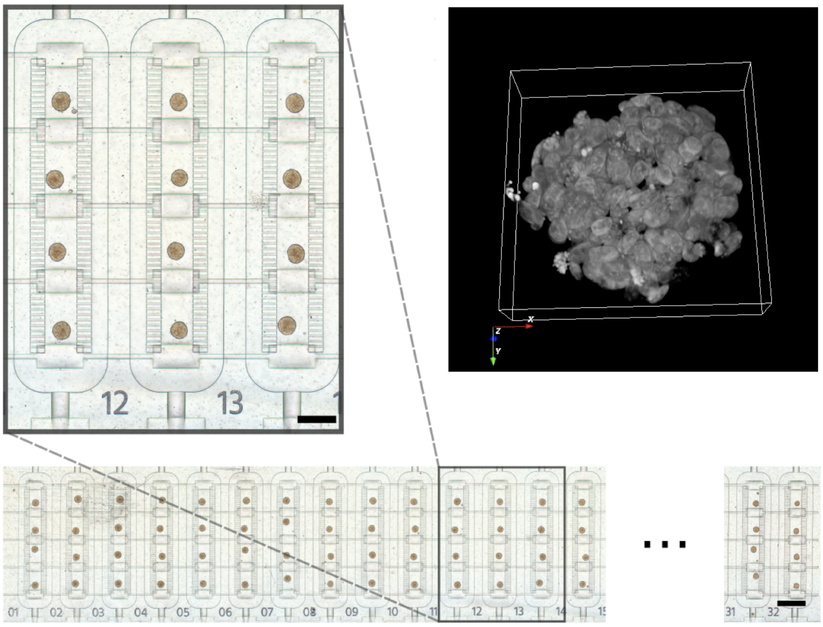


Figure 1.2: Visualization of a microfluidic platform for 3D cell culture experiments. 128 cell cultures can be grown in parallel in a controlled microenvironement with 32 different conditions. Four cell cultures share the same condition, which can be controlled automatically with a valve system. This allows for a parallelization of experiments, for example in the application of drug testing or in stem cell research [4]. In the upper right corner, one fixed cell culture with a fluorescent nuclear marker imaged with a confocal microscope is visualized. Scale bars: 500µm (top), 1000µm (bottom).

Confocal fluorescence microscopy can not overcome the downsides of fluorescent imaging, however it is an excellent technique for quantitative imaging of cells and tissues. Data acquired with a confocal microscope has mainly been used in this thesis (see Figure 1.1). One main advantage over traditional widefield or epifluorescence microscopes is the ability to generate high-contrast images through optical sectioning. By blocking out-of-focus light, an optical sectioning through a specimen can be performed, allowing for quantification of fluorescent signals with very high spatial precision [21]. In contrast, traditional widefield microscopes acquire the image as a superposition of sharp features originating from the focal plane and blurry features from outside of the focus [21]. In a confocal microscope, a laser beam is focused into a specimen, its cone of illumination excites fluorescent molecules. These emit light of a longer wavelength than the wavelength of the excitation laser, which is collected by the objective lens. The collected light is focused by another lens through a pinhole, ensuring that only fluorescence that originates at the focal point is captured by the camera. Thereby, fluorescence emission from outside the focal plane is blocked [21]. Because the microscope collects fluorescence signals from only one focal point at a time, the specimen is scanned using mirrors that sweep the laser beam across the specimen, generating an image pixel by pixel [21]. Scanning times are around one second for one slice containing 1014x1024 pixels, for 3D imaging this can easily result in minutes of imaging time per volume [4].

In modern biological experiment designs, researchers seek to reduce the complexity of biological processes with *in-vitro* models that mimic the architecture of the tissue niche, promoting cell-to-cell and cell-to-matrix interactions [4]. Thereby, *in-vitro* disease models can be obtained and regenerative therapies explored so that drug screens requiring animal experiments can be replaced [22]. Microfluidic chip technologies have emerged to provide controllable environments by automating fluid programs where cell cultures can be grown (see figure 1.2). Microfluidic platforms combined with confocal microscopy therefore offer researchers a way to investigate cellular processes in a controllable microenvironment, mimicking biological processes observable with high temporal and spatial resolution. Parallelization of these experiments results in large amounts of acquired images that need to be processed. Phenotyping or tracking cells in 3D cultures in high-throughput experiments is a challenge because cells cultured in 3D adopt higher density configurations than in 2D cultures and cell morphologies in 3D cultures are more heterogeneous and less recognizable than in 2D cultures [4].

1.2. Machine learning

Machine learning is a computational technique, that enables algorithms to adapt to some degree automatically. Thereby they learn automatically how to solve a task. In 1997, Tom Mitchell defined machine learning: "A computer program is said to learn from experience E with respect to some class of tasks T and performance measure P if its performance at tasks in T, as measured by P, improves with experience E" [23]. The experience E is information that is contained in the data-set a machine learning algorithm is trained on. The task T is defined by the user, that can be for example classifying single cells or segmenting an image. Depending on the task different algorithms are to be applied. The performance of the algorithm to solve task T is measured by a performance measure P that can be any metric. Common uses are for example the mean-squared error to evaluate pixel-value predictions, or the cross entropy for classification tasks. However machine learning algorithms share the ability to learn and adapt without following explicit instructions by drawing inferences from patterns in data. For task T, we use experience E to train a mapping f from the models inputs X, that is a subset of the experience E to the ground truth Y, that is another subset of the experience E [24]. The inputs X and the groundtruth Y is provided by the user to the model, both must not have an overlay, else the model will only learn trivial solutions. The experience E can be provided by N data points, which are pairs of $x_i \in X$ and $y_i \in Y, i = \{1, ..., N\}$ in the scenario of supervised training, where y_i are ground truth labels. Or in the scenario of unsupervised training, where no ground truth labels are available, only by $x_i \in X \; \forall i \in N$, this will be discussed in more detail in the following section 1.4. During training, model parameters are iteratively modified, improving model performance P with experience E. A loss function is employed to quantify the performance P of model parameters during training.

Depending on the machine learning task, the cost function can vary from mean absolute error or mean squared error to more specific functions, such as the dice loss [25], which is often used for object segmentation (see section 1.3.3). Choosing a suitable loss function is an influential decision as it determines to which optimum the model converges [26]. During training, a validation set is used to monitor if and how effective the optimization procedure converges the model parameters towards a minimum of the cost function. After training the model is evaluated on a held-out test set.

1.3. Classification

Classification specifies the task to assign a class label to each input, for example assigning a label to each image in a data-set [24, 27]. The output space C in classification tasks is a set of k mutually exclusive, unordered labels or classes $C = \{0, ..., k-1\}$. For every data point x_i in the data-set, there is a label y_i and a mapping $f(X) = Y, \forall x_i \in X, y_i \in Y, i = \{1, ..., N\}$, where the number of classes is always smaller than the number of input data points N. Typically, each class label is represented by an integer. Tasks with only two classes $y_i \in \{0, 1\}$, are called binary classification [24, 27]. In a classification task the output is a class label, and the ground truth typically a positive integer, as such requiring a classification loss, such as the categorical cross entropy (see section 1.3.3).

1.3.1. Regression

If continuous real-valued numbers $y_i \in \mathbb{R}\ \forall i \in N$ are predicted instead of discrete class labels, the task is called regression [24, 27–29]. The label can be a single real-valued number or may be of arbitrary dimensions, for example pixels in an image. In a regression task the output can have any dimensions, for example be a single number, or pixel values in an image. For all regression tasks the model's output and the groundtruth are both continuous floating point numbers, requiring a regression loss, such as the mean squared error (see section 1.3.3).

1.3.2. Segmentation

Image segmentation is the task of classifying each pixel in an image and thereby partitioning regions of an image, with the goal to identify pixels belonging to objects [24, 27]. The inputs x_i and outputs $y_i \in \mathbb{N}$, typically have the same spatial dimensions. Machine learning researchers differentiate between semantic segmentation and instance segmentation [24, 27]. The goal of semantic segmentation is to classify each pixel, while in instance segmentation objects of the same class are first individual objects in an image are identified and then object of the same class are segmented [24]. In a segmentation task the output typically has the same dimension as the input, as for every input pixel a label is predicted. The ground truth for this task is typically a pixel map with positive integers as class labels, allowing for the application of segmentation loss functions, such as the dice loss (see section 1.3.3).

1.3.3. Optimization and metrics

The choice of the cost function is crucial as the performance measure T of a machine learning model [26]. The cost function can be any differentiable and convex mathematical function. This metric is also used to evaluate the performance of a trained model on an unseen test set. The most important metrics to train and evaluate machine learning models are listed in the following. For a model f with parameters θ, the mapping $f(x_i, \theta)$ describes the model prediction, while y_i denotes the ground truth.

$$Meanabsoluteerror(MAE) = \frac{\sum[|f(x_i, \theta) - y_i|]}{N} \tag{1.1}$$

The mean absolute error is often used in regression tasks.

$$Meansquarederror(MSE) = \frac{\sum[f(x_i, \theta) - y_i]^2}{N} \tag{1.2}$$

The mean squared error is often used in regression tasks.

$$Binarycrossentropy(BCE) = \frac{\sum[y_i * \log(f(x_i, \theta)) + (1 - y_i) * \log(1 - f(x_i, \theta)]}{N} \tag{1.3}$$

The BCE is often used in binary classification or binary segmentation tasks.

$$Categoricalcrossentropy(CCE) = -\sum[y_i * \log(f(x_i, \theta))] \tag{1.4}$$

The CCE is often used in multi-class classification or multi-class segmentation tasks.

$$Diceloss = \frac{2 * \sum y_i * f(x_i, \theta)}{(\sum y_i)^2 + (\sum f(x_i, \theta))^2} \tag{1.5}$$

The Dice loss is often used in image segmentation tasks. To define *precision*, we first need to define *true positive*, which is the outcome when a model correctly predicts the positive class. Accordingly, a *true negative* is an outcome where the model correctly predicts the negative class. A $false$ *positive* is when the model incorrectly predicts the positive class, a $false$ *negative* is an outcome where the model incorrectly predicts the negative class [24].

$$Precision = \frac{true\,positive}{true\,positive + true\,negative} \tag{1.6}$$

$$Recall = \frac{true\,positive}{true\,positive + false\,negative} \tag{1.7}$$

$$F1Score = \frac{precision * recall}{precision + recall} \tag{1.8}$$

$$Accuracy = \frac{true\,positive + true\,negative}{N} \tag{1.9}$$

1.4. Learning strategies

For a given data-set D, containing N data points x_i and possibly the corresponding ground truth labels $y_i \forall i \in N$, typically a fraction of the data points is removed from the training process as a separate test set. On this test set the model is evaluated after the training. The remaining data-set is called the train set. The test set must not contain data also contained in the train set, but its distribution should reflect the distribution of the whole data-set. From the train set, typically 20% of the data is split into a validation set that is used to monitor the performance of the model during training. The model is then optimized to maximise the performance measure P, which depends on the task.

1.4.1. Supervised learning

Supervised learning is the most common form of machine learning. For task T, a mapping f is learned $f(x_i) = y_i$ from the inputs $x_i \in X$ to the outputs $y_i \in Y$, with the goal to minimize the cost function $|f(x) - y|$ where $y_i \in Y$ is also called ground truth [24]. The experience E is provided by N data points, i.e. pairs of x_i and y_i $\forall i \in N$.

1.4.2. Adversarial learning

Adversarial training is a subcategory of supervised training, where a second model, the adversarial model, is trained with the goal to improve the performance of the original model [30]. The adversarial model can for example be a discriminator, such as a classification model, which is trained to differentiate between samples from the ground truth data-set and the predictions of the original model. The two models are typically trained in an alternating fashion, which can be interpreted as a contest between the two models, where they iteratively challenge each other and thereby improve their performance. During training, the adversarial model improves its classification accuracy in each training cycle, while the original model is optimized generate plausible examples and thereby fool the adverserial model.

1.4.3. Unsupervised learning

In unsupervised learning, a model is trained without ground truth Y. These algorithms can also be used for clustering [24]. As no ground truth is needed, unsupervised learning is especially attractive for tasks where the collection of labeled data-sets is time consuming or expensive, which is the case for many biomedical applications. In the biomedical field data labeling often requires a trained expert to manually annotate a data-set. Autoencoders (see section 1.6.5) are a typical application of self-supervised training, where the input data is encoded to a lower dimension and then decoded with the aim to recover the input data. The performance P of the Autoencoder is measured using the reconstruction error $|f(x) - x|$. Self-supervised learning can be used as a pre-training of a model, which can later be fine-tuned with supervised learning using a small set of labeled data.

1.4.4. Self-supervised learning

One subclass of unsupervised learning is self-supervised learning, where a supervised task is created to train a model on unlabelled data using implicit labels from the data-set [31].

1.5. Deep learning

As a branch of machine learning, deep learning has emerged in the last decade. Deep learning models are artificial neural networks that tries to resemble the human nervous system or the human brain structure [32]. It is typically built from input layers, output layers, and hidden layers [24]. While each layer is a mathematical function. Hidden layers allow for modeling complex nonlinear relationships, they are called hidden, because the true workings are only indirectly accessible. As opposed to machine learning models, deep learning methods are characterized by several hidden layers performing nonlinear transformations. This enables them to generate more abstract and complex mappings between the inputs x and outputs $f(x)$ [24]. The key idea behind deep learning is to estimate a function,

$$f(x) = w * x + b \tag{1.10}$$

which maps the input x to the output y, with the objective to minimize the cost function

$$|f(x) - y| \tag{1.11}$$

The flexibility of the model can be increased by replacing x with a feature transformation, that is a function $\phi(x)$, sometimes called basis function expansion. For example a polynomial function could be used.

$$f(x) = w * \phi(x) + b \tag{1.12}$$

To overcome the limit of having specific feature transformations we can endow the feature extractor with its own parameters θ

$$f(x, \theta) = w * \phi(x, \theta) + b \tag{1.13}$$

We can stack these functions to obtain more complex functions, each function containing its own parameters:

$$f(x, \theta) = (f_L(f_{L-1}(...(f_1(x)...))) \tag{1.14}$$

where $f_L(x)$ is the function at layer L. This allows deep multi-layer networks to have many levels of non-linearities enabling them to compactly represent highly non-linear and highly-varying functions.

1.5.1. Optimization for deep learning models

To adjust model parameters, the backpropagation algorithm is used [24]. Typically the parameters of the model are initialized randomly or drawn from a distribution, such as the normal distribution [24]. After each training step, the loss between the output of the model and the ground truth is calculated using the cost function (see section 1.3.3) and is then propagated backwards through the model, adjusting the parameters of the model for improved predictions, thereby learning the best model parameters [24]. This can be reformulated as searching the minimum of the cost function, where in each training step the minimum is approached iteratively [24]. The direction in which the parameters of the model are optimized is calculated by the gradient of the cost function. This algorithm is called gradient descent [24, 33]. The gradient descent algorithm calculates the direction using gradient at the current position, then scales it by the learning rate and subtracts the obtained value from the current position to make an optimization step. The learning rate is a parameter that is multiplied with the gradient to regulate amount of change that is applied to the parameters of the model. The learning rate is a crucial parameter for optimizing a neural network. If the learning rate is chosen too high, the model will converge fast, but might overshoot the minimum; if the learning rate is chosen too low it will take long to converge. Often the learning rate and automatically adapted during training either by a predefined schedule or if the loss has not decreased for a few iterations. One widely used extension to the gradient descent algorithm is called Adam [34], its main improvement is the application of per-parameter learning rates by calculating the average of the first and second moments of the gradients [34].

1.5.2. Layers in a neural network

Deep learning models are typically built from multiple layers [24]. While new layers are constantly developed, some have found wide acceptance in the machine learning community and are commonly used, so in this thesis. One key technique in deep learning models for image analysis are convolutional layers, which are inspired by the human visual system's structure [24]. The main challenge when constructing a deep neural network is designing the sequential order of layers, also referred to as the network architecture. As not all deep learning architectures perform equally well on different tasks, developing and benchmarking architectures is an ongoing process in machine learning research. Generally, the more layers a model has, the better it can abstract the input data to higher order patterns [24]. A model with more layers has more parameters and is therefore more difficult to train and is increasingly prone to overfitting if the data-set is not sufficiently large and diverse. A model which perfectly fits the training data, but which is too complex, is said to suffer from overfitting, it often fails to generalize to data not contained in the training set [24]. Large models often suffer from vanishing backpropagation gradients, when gradients become smaller for layers closer to the input during backpropagation through the network [35]. Small models are fast to train but might lack the ability to capture the complexity required to solve a task. Before describing architectures that are routinely used by the machine learning community in biomedical data analysis, I will describe the layers used to build these architectures. Frameworks such as TensorFlow [36] or PyTorch [37] have been developed by Google and Meta, respectively, where frequently used layers are implemented, easing the development of deep learning methods for researchers.

- **Convolutional layer** extract feature maps from the input data. They are inspired by biology, such as the connectivity pattern between neurons in a human brain and by the organization of the visual cortex [24]. The parameters of each convolutional layer p are a set of learnable filters or kernels, with a small receptive field. The basic idea is to divide the input into overlapping patches, and to compare each patch with a set of small weight matrices, filters, or kernels. This can be thought of as a form of template matching [24]. The model will learn the best templates during training. Because the templates are small (often just 3x3 or 5x5, in the 2 dimensional case), the number of model parameters is significantly reduced as compared to larger templates [24]. Upon receiving the input, a filter is moved across the dimensions of the layers' input, while the dot product between the filter entries and the input is calculated. During training the model learns to optimize the convolutional layers parameters so that filters are activated when specific features in the input are detected [24]. Because convolutions are used for the template matching, instead of dot products, the model will be translationally invariant. This is useful for tasks such as image classification, where the goal is to classify if an object is present, regardless of its location [24].

- **Normalization layer** normalize the input of a layer. They are important when building deep neural models as they standardize the statistics of the hidden layers and can prevent the gradient from vanishing or exploding and thereby impede the models training. Most commonly a normalization by the mean and standard deviation is used [24]. Batch normalization layers standardize the whole batch of the layers input data [24]. This layer does not learn any parameters.

- **Pooling layer** reduce the dimensions of the layers input and thereby allow to down-sample the input in a convolutional neural network, while leaving the channel size untouched [24]. They combine inputs from a fixed set of inputs by calculating their mean or maximum. A common example is a max-pooling layer, where the maximum of a set of input features is calculated and passed forward. As opposed to convolutional layers, which preserve information about the location of the input features, pooling layers process the information invariant from its location, as they pass the same response no matter where in their receptive field the information occurs [24]. This layer does not learn any parameters.

- **Upsampling layer** perform an upsampling to the input data by extrapolating to nearest neighbours via linear, bilinear or trilinear interpolation in a convolutional neural network. They can be used to increase the dimensions of the input and to perform the inverse operation to the pooling layer [24]. This layer does not learn any parameters.

- **Fully connected layer** process information contained in the whole input tensor. They connect every neuron of a previous layer to every neuron in the next layer and therefore have a large number of parameters. They are most often used to bundle all information allowing for high-level reasoning. Their activation can be calculated by a matrix multiplication with a bias offset [24].

- **Dropout layer** randomly erase parts of the input. These are often utilized for regularization, forcing the model to perform well even if some of the input is missing [38, 39]. This layer does not learn any parameters.

- Activation functions map the input x to an output $f(x)$ via a fixed mathematical function [24]. The most common activation functions used in deep neural models today are listed in the following and shown in figure 1.3.

  - Binary

$$\begin{aligned} f(x) &= 0; \text{ if } x \leq 0, \\ f(x) &= 1; \text{ if } x \geq 0 \end{aligned} \tag{1.15}$$

  - Linear

$$\begin{gathered} f(x) = \alpha * x \\ \text{with } \alpha \in \mathbb{R} \end{gathered} \tag{1.16}$$

  - Sigmoid

$$f(x) = \frac{1}{1+e^{x}} \tag{1.17}$$

  - Hyperbolic tangent

$$f(x) = \frac{e^{x} - e^{-x}}{e^{x} + e^{-x}} \tag{1.18}$$

  - Rectified linear unit (ReLU)

$$\begin{aligned} f(x) &= 0; \text{ if } x \leq 0, \\ f(x) &= x; \text{ if } x \geq 0 \end{aligned} \tag{1.19}$$

  - Leaky rectified linear unit (Leaky ReLU)

$$\begin{aligned} f(x) = \lambda * x; \text{ if } x \leq 0 \\ f(x) = x; \text{ if } x \geq 0 \\ \text{with } \lambda \in \mathbb{R} \text{ and typically } \lambda \leq 1 \end{aligned} \tag{1.20}$$

In practice, leaky rectified linear functions are often used, which allow a small non-zero gradient for input values smaller than zero [24].

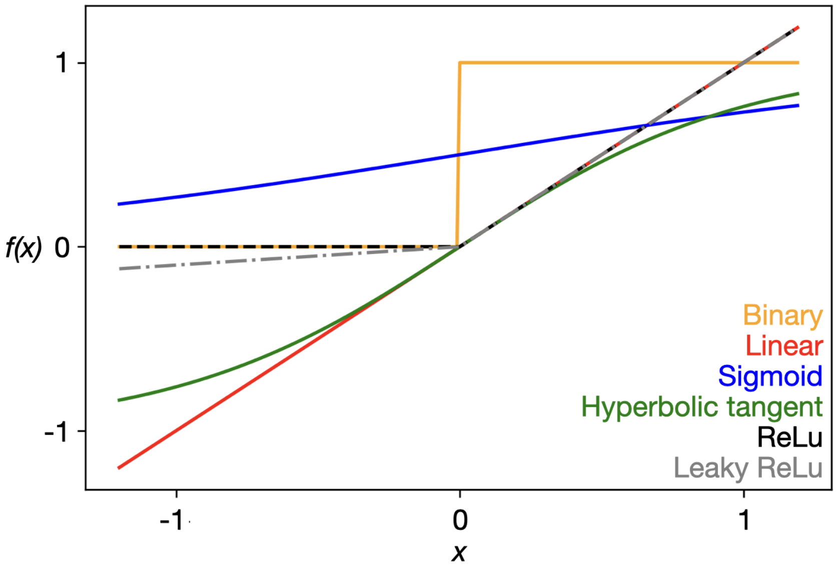


Figure 1.3: An activation function in a neural network transforms the weighted sum of the input x to an output $f(x)$ for each node in the network. Using nonlinear activation functions allows the model to learn complex patterns. Depending on the desired task some activation functions have proven more suitable than others [24, 32]

1.5.3. Techniques to improve predictive performance of deep neural models

Performance of a deep neural model can be improved by either making the training more efficient, such as using transfer learning, or by avoiding pitfalls that hinder the model to perform well, such as overfitting. Overfitting is a common pitfall when training neural models, meaning that a model performs well on training data but fails to generalize to unseen data [24, 40]. Such a situation can be identified if the training loss steadily decreases, while the validation loss increases. Multiple techniques have been developed to prevent overfitting, summarized under the term regularization. Regularization techniques include data augmentation, dropout and early stopping, which will be described in the following.

- **Data augmentation** can be applied to enrich the data-set by spatial or color augmentations and thereby increase ability of the model to generalize. Spatial augmentations include image rotation, translation, mirroring or zooming. Color augmentations include brightness changes, contrast changes, or blurring of the images, addition of noise and image normalization [41–43].

- **Dropout layers** randomly deactivate a fraction of the neurons of the model at each training step. This can prevent overfitting, increase the ability of the model to generalize and improve the predictions robustness [44]. During interference, dropout layers can be used for model averaging and uncertainty quantification [45].

- **Early stopping** is a method to measure when a model has trained enough and might start overfitting. The predictive of the model performance is measured on the validation data-set during training. If the validation error no longer decreases for a certain number of training steps the training will be interrupted automatically saving computational resources [33].

- **Transfer learning** uses the parameters of trained models to initialize a the parameters of a second model, allowing for faster convergence of the model and superior predictions [6, 46]. The weights for model initialization can for example be obtained by training a model on another labeled data-set in a supervised fashion. Often, weights for commonly used models are published and can directly be applied on new tasks.

- **Batch computing** is used to increase the efficiency of the optimization by consolidating a number of input data points, called batch and processing them simultaneously. This increases the robustness of the gradient during training [24].

- **GPU computing** is using a graphics processing unit (GPU), which can process tasks magnitudes faster than the central processing unit (CPU) due to its parallel processing capabilities and thereby speed up training and interference using deep learning models [47]. Today's deep learning frameworks are optimized to be used with a GPU.

1.5.4. Uncertainty quantification in deep learning

Assessing the quality of the prediction of a neural network is not straightforward. Predictions of a deep learning model can be unreliable or wrong if the model fails. Model failures can be caused if the input not inside the training distribution or if the input is corrupted. Uncertainty quantification is one method that measures the uncertainty of a model for each prediction, thereby the robustness of a model can be assessed, which adds a new level of insights and interpretability to model predictions. Bayesian statistics allow us to assess uncertainty in deep learning models, considering both both data and prior knowledge. One of the presumptions is that model parameters are not fixed weights but distributions. Ensembling is one method to obtain model weigh distributions, however it requires training a large number of models to obtain prediction distributions. However, Bayesian inference can be also approximated in deep Gaussian processes by Monte Carlo dropout [45], which is an efficient way to measure the uncertainty of predictions of deep learning models. It does not require any retraining of the model, only multiple inferences with Monte Carlo dropout. Therefore Monte Carlo dropout can be utilized for uncertainty quantification, requiring dropout layers (see section 1.5.2) to be added to the model [45]. If these are activated during inference, which is an easy approach to perform model averaging. Several forward passes of the same data point are performed with randomly changing deactivated neurons, thereby a distribution of predictions can be obtained from which the uncertainty of the predictions of a model can be calculated [45].

1.5.5. Topological loss function

The loss function (see section 1.3.3) is a crucial parameter when optimizing a deep learning model to predict the desired output [26]. Established geometrical loss functions such as the MSE, dice loss or BCE (see section 1.3.3) only capture errors made between pixels or voxels and do not consider contextual information. Topological data analysis, by contrast, provides contextual information by characterizing data in terms of pixel or voxel connectivity (or even generic point cloud connectivity).

Topology is a field in mathematics concerned with the properties of a geometric object that are preserved under continuous deformations. Continuous deformation include stretching, twisting, crumpling, and bending the object [48]. During these deformations no holes in the object are opened or closed, nothing is teared, glued or passed thought itself.

A topological space is a set endowed with a structure, called a topology, that allows for the definition of continuous deformation of subspaces, and, more generally speaking, all kinds of continuity. Metric spaces, such as the euclidean space are examples of a topological space, as a topology is defined by a distance or metric defines a topology. Deformations considered in topology are homeomorphisms and homotopies. A property invariant under such deformations is a topological property. Examples of topological properties are: the dimension, that allows for the differentiation between a line and a surface and the compactness, that allows for the differentiation between a line and a circle, and the conceitedness that allows for the distinguishing between a circle from two non-intersecting circles. These topological features provide a different and enriching view on the differences between the data points processed in machine learning.

These topological features primarily characterize data primarily in terms of connectivity, thus giving a different perspective on data that can be utilize to compute a loss term that penalizes differences in connectivity patterns. The resulting topological features can be utilized to construct a loss function that compares connectivity patterns instead of geometrical information.

Topologial features

To extract topological features from images or volumes a cubical complex is used to represent the data. Cubical complexes are related to meshes but use squares instead of triangles as their building blocks. Meshes are one possibility to describe volumetric data with a grid fitted to their surface. Cubical complexes can represent image data by containing individual pixel or voxel as vertices, and connectivity information about their neighborhoods via edges, squares and higher-dimensional counterparts [3, 49]. In 0 dimensions (0D) topological features are connected components, in 1D cycles, and in 2D voids. A topological dimension in space X has dimension n if n is the smallest natural number such that for every open covering there is $(U_i)_{i \in I}$ a finer open covering $(V_j)_{j \in J}$ such that every point from X in at most $n + 1$ of the sets V_j lies. If there is no such n, then X of infinite dimension. The dimension of X is given by $\dim(X)$. Since there is a whole set of other dimension terms, one speaks more precisely of the Lebesgean covering dimension [50, 51].

Persistent Homology

Based on that representation multi-dimensional topological features can be calculated [52] with persistent homology as follows. In the setting of image segmentation a probability $f(x)$ for each pixel or voxel x is predicted by the model $f\colon \mathcal{V} \to \mathbb{R}$, indicating if a pixel or voxel belongs to the segmented object. For different probability thresholds $\tau_1 \geq \ldots \geq \tau_m$, $\tau_i \in \mathbb{R}$ applied to the predicted values of a model $f(x)$ different segmentation can be obtained. Therefore this results in different cubical complexes $C^{(\tau)} := \{x \in \mathcal{V} \mid f(x) \geq \tau\}$ that can be calculated, leading to different sets of topological features. The cubical complexes change at finite numbers, since pixel values are finite, leading to a sequence of nested cubical complexes $\emptyset \subseteq C^{(\tau_1)} \subseteq C^{(\tau_2)} \subseteq \cdots \subseteq C^{(\tau_m)} = \mathcal{V}$, which is called a *superlevel set filtration* [3]. We use a method called Persistent homology that can track topological features across all cubical complexes in the filtration and relate each feature to a tuple (τ_i, τ_j), with $\tau_i \geq \tau_j$, indicating when the feature first appeared in the cubical complex, called 'birth', and when it disappears, called 'death'. The tuples extracted from the cubical complex k-dimensional features, with $0 \leq k \leq d$, are stored in the kth persistence diagram $\mathcal{D}_f^{(k)}$, forming a multi-scale shape descriptor of all topological features in an image. For a tuple (τ_i, τ_j) in a persistence diagram, the sum of all persistence values is known as the degree-p total persistence, it is defined as: $\mathrm{Pers}_p(\mathcal{D}_f) := \sum_{(\tau_i,\tau_j)\in\mathcal{D}_f} \left|\mathrm{Pers}(\tau_i, \tau_j)^p\right|$ [3]. Using the Wasserstein distance, a metric used in optimal transport, two persistence diagrams $\mathcal{D}$ and $\mathcal{D}'$ can be compared [3, 53]. The pth Wasserstein distance is defined as:

$$\mathrm{W}_p(\mathcal{D}, \mathcal{D}') := \left(\inf_{\eta\colon \mathcal{D}\to\mathcal{D}'} \sum_{x\in\mathcal{D}} \|x - \eta(x)\|_\infty^p \right)^{\frac{1}{p}}, \tag{1.21}$$

where $\eta(\cdot)$ describes a bijection. Because $\mathcal{D}$ and $\mathcal{D}'$ most likely have different cardinalities, we assume they contain an infinite number of tuples (τ, τ) with zero persistence to calculate suitable bijections. Therefore suitable bijections $\eta(\cdot)$ can always be calculated and with optimal transport, solving equation 1.21 is feasible and the equation is well-defined. Although topological features are discrete, persistent homology allows for calculating gradients with respect to parameters of the likelihood function f, allowing for automatic differentiation schemes [54–56]. Optimization algorithms converge for a wide class of persistence-based algorithms, as proven by Carrière et al. [57], enabling topology-based optimization loss functions.

With that the ground truth y_i and the prediction of the model $f(x_i)$ we can calculate a topology-aware loss function by:

$$\mathcal{L}_{\mathrm{T}}(y_i, f', p) := \sum_{i=0}^{d} \mathrm{W}_p\left(\mathcal{D}_{y_i}^{(j)}, \mathcal{D}_{f(x_i)}^{(j)}\right) + \mathrm{Pers}\left(\mathcal{D}_{f(x_i)}^{(j)}\right). \tag{1.22}$$

The first part of equation 1.22 regularizes the model to minimize the Wasserstein distance between topological features extracted from y_i and $f(x_i)$. The second part regularizes the model to minimize overall topological activity, thereby decreasing the noise in the reconstruction [3].

1.6. Machine and deep learning models

Five machine and deep learning models have been used in this thesis which are introduced in the following sections: random forest, residual networks, Mask R-CNN, autoencoders, and U-Net.

Random forests are machine learning models that can be used for feature based classification and regression, that is also the exact application of them in this thesis. Residual networks, Mask R-CNN, autoencoders and the U-Net are deep learning networks that are applied on images. Residual networks are designed to classify images, in this thesis they have been used to classify microscopy images. The Mask R-CNN is an object detection network that detects instances in an image and then classifies and segments them. In this thesis it was used to segment images. In this thesis autoencoders have found application on the task of 3D shape prediction from 2D images, where I have stacked a 2D encoder with a 3D decoder. The U-Net is a semantic segmentation network, as the Mask R-CNN it has been used in this thesis to segment microscopy images and as a regression model to predict fluorescent stainings from bright-field images. Before introducing the models in detail an essential part of every machine learning project, the data pre-processing is elaborated.

1.6.1. Data pre-processing for deep learning

Preparing a data-set be one of the most difficult and influential steps for any machine learning project. Labels for supervised training need to as be error free as possible, because errors might be adapted by the model [58]. While the data preparation process is highly data-set specific, there are a few commonalities that are mentioned in this section. Once the aim of the machine learning project is defined, the right model can be selected and the data we assume is most likely to lead to a successful project outcome can be collected. Often, as a second step the data needs to be reformatted, as some data types such as specific microscopy image files might not be readable by the machine learning framework. This second step also includes data cleaning, as often data-sets contain errors, noise or meaningless data points, preventing the model from learning on false or irrelevant data. After that, properties that are irrelevant to the machine learning task, such as large background spaces in an image, should be identified and removed, to shrink the data-set size and improve learning speed and accuracy. When working with images, this step can include a rescaling step of the pixel or voxel size and patching or cropping images to obtain the desired dimensions for the task at hand. Machine learning models perform best on normalized data-sets, where data points are on the same scale [24, 59]. Standard techniques are normalization with a minimum and maximum, or a mean and standard deviation normalization.

1.6.2. Random forest

There are numerous published machine learning algorithms, the Random forest algorithm is the only one used in this thesis. Random Forest is a supervised machine learning algorithm, constructed from an ensemble of decision trees [32]. Decision trees consist of a set of nested binary decision rules that are inferred from the data points $x_i i \in N$. At each node of a tree, the input vector of the node is compared to a threshold value and thereafter passed down to one of two subsequent branches. At the end of this process, the model predicts an output for any input that falls into that part of the input space. To fit a tree to a data-set, the tree is grown iteratively by one node per step. Thereby, the loss function is used to minimize the entropy or to maximize the information gain. In a random forest algorithm, an ensemble of decision trees is constructed, each with a randomly chosen subsets of the data and with randomly chosen subsets of features, resulting in a sufficiently diverse base model [24, 60]. The random forest prediction can be obtained by averaging of all decision trees predictions (see figure 1.4).

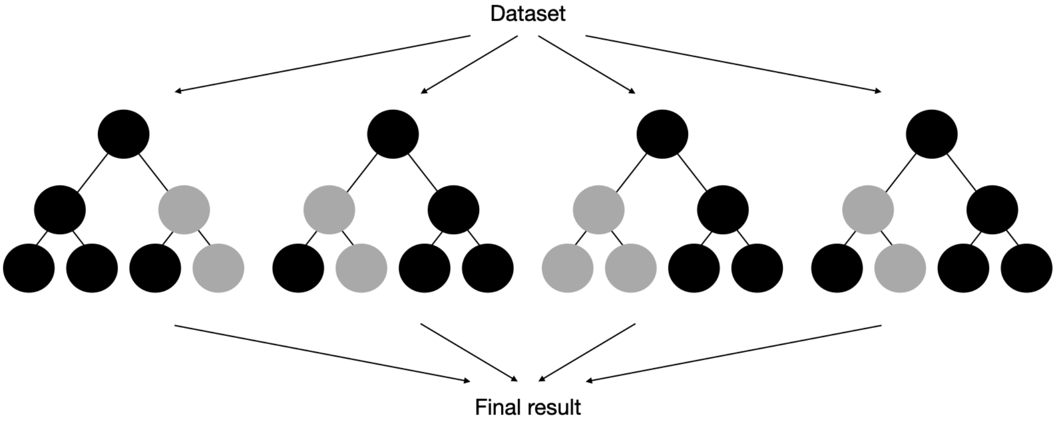


Figure 1.4: Schematic visualization of a Random forest. An ensemble of decision trees is constructed, each with a randomly chosen subset of the data-set and subset of features.

1.6.3. Residual networks

Residual neural networks (ResNets) are a class of deep learning models that can be used for image classification. As deep learning models are constructed with greater depth to increase their performance, they become increasingly difficult to train, as the back-propagated gradients to earlier layers becomes infinitely small due to repeated multiplications [24, 35]. This can lead to a performance saturation or even degrading when models nested. The ResNet architecture tackles this vanishing gradient problem by introducing ‘identity shortcut connections’, which skip one or more layers [35]. These stacked layers allow gradients to flow through shortcut connections to any other earlier layer. The key idea is to replace the layer $f_l(x)$ with the output x_{l+1}

$$x_{l+1} = f_l(x) \tag{1.23}$$

with

$$x_{l+1} = x_l + f_l(x). \tag{1.24}$$

This is called the residual block, since f_l only needs to learn the residual between input x_{l+1} and output $x_l + f_l(x)$ of this layer. As gradients in a residual model can flow directly from the output to earlier layers, these are easier to train, while having the same number of parameters as a model without residual connections [24]. Today, ResNet and its variants are often used for classification tasks in biomedical imaging [61].

1.6.4. Mask R-CNN

The Mask R-CNN is a Region Based Convolutional Neural Network (R-CNN) that can be used for object detection and instance segmentation [35]. It is composed of a multi-stage process, starting with a region proposal network that proposes candidate regions for object bounding boxes. In a second stage, features from these candidate bounding boxes are extracted and a bounding-box regression and object classification is performed. Additionally, Mask R-CNN segments each instance in a bounding box (see figure 1.5). Thereby, multiple objects in an image can be detected, classified and segmented at once. As backbone, Mask R-CNN uses a ResNet architecture (see section 1.6.3) that can be pre-trained e.g. on ImageNet (see section 1.5.3), a data-set containing over 14 million hand annotated images [35, 62]. During training, a multi-task loss is calculated as the sum of the object detection, classification, and segmentation loss.

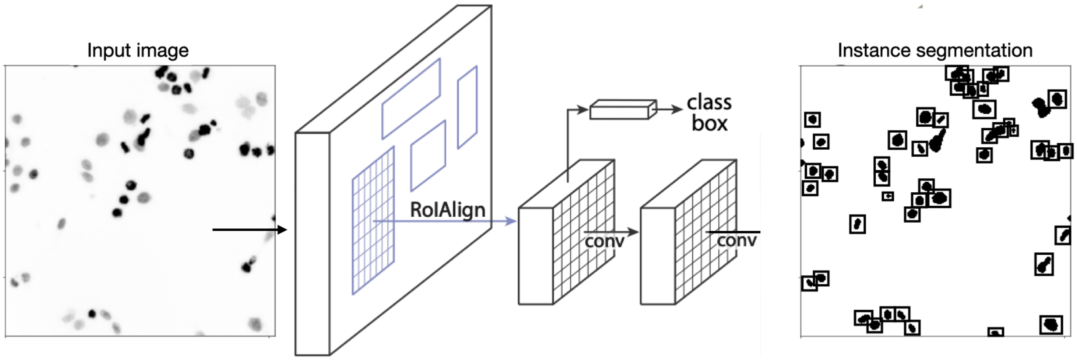


Figure 1.5: Schematic visualization of the Mask R-CNN model architecture. In each input image objects are detected, aligning one Regions of interest (RoI) for each detected object. In the next step each object in each RoI is classified and segmented using convolutional layer.

1.6.5. Autoencoder

An autoencoder is a deep learning model that learns an efficient embedding of data points and then regenerates the predictions from embeddings (see figure 1.6). Autoencoders are a classical application of self-supervised training (see section 1.4.4). Autoencoders have been applied for many different tasks, such as denoising, feature encoding, abnormality detection, or self-supervised pre-training [24, 63]. An autoencoer is constructed of two modules, the encoder f_e and the decoder f_d, which are often symmetric to each other. The encoder is a network that reduces the input dimensions and compresses the input data into an encoded representation. This encoded representation is contained in the so-called bottleneck layer, which links the encoder with the decoder. The decoder reconstructs the output from the encoded representation to match the input data [24, 63]. Encoder and decoder architectures can be composed of consecutive convolutional, normalization and downsampling or respectively upsampling layers (see section 1.5.2). The reconstruction function has the form:

$$a(x) = f_d(f_e(x)). \tag{1.25}$$

Often a reconstruction loss

$$L(\theta) = ||a(x) - x|| \tag{1.26}$$

between the input and output of the model is used to train autoencoders. Therefore, they can be used for self-supervised training as no ground truth labels are required for training.

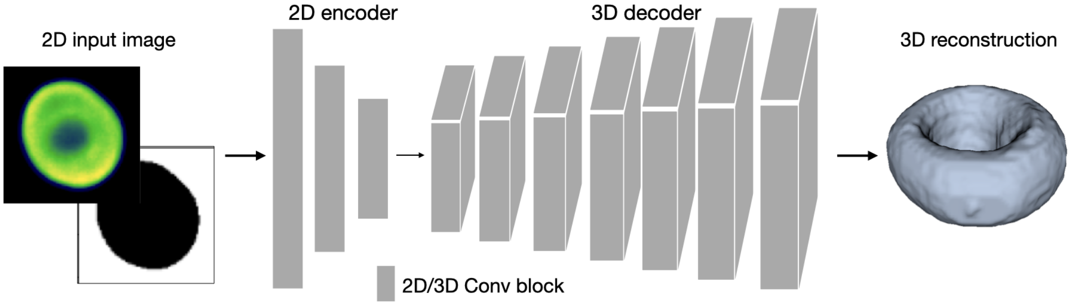


Figure 1.6: Schematic visualization of an autoencoder model architecture. In this example each 2D input image is encoded into a latent space representation, before it is decoded again to a 3D single cell shape. For this 2D and 3D convolutional layers are utilized.

1.6.6. U-Net

A U-Net is a deep neural network with primary applications in semantic segmentation and pixel-wise regression of 2D and 3D images [64, 65]. Similar to an autoencoder, it consists of downsampling and upsampling steps. For each downsampling step the number of features per layer is doubled while the dimension is halved and vice versa in the upsampling steps [64, 65]. This symmetry yields the U-Net's name (see figure 1.7). So called skip connections between downsampling and upsampling layers enable the preservation of high-frequency information and are concatenated to the respective upsampling layers. Each layer consists of two convolutional layers, followed by an activation function and a dropout layer (see section 1.5.2). For downsampling steps, a pooling layer is added and for upsampling steps an upsampling layer is used.

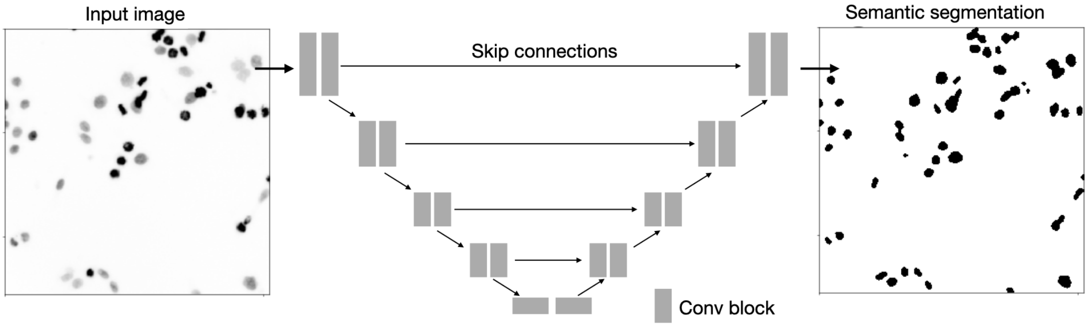


Figure 1.7: Schematic visualization of a U-Net model architecture. U-Nets downsample the input image using stacked convolutional layers, and then upsample the encoding again. To preserve high-frequency information skip connections are used to concatenate the output of the downsample layers to the respective upsample layers.

1.7. Machine learning applications in microscopic imaging

Recent progress in microscopic image analysis has been driven by machine learning [12, 24, 66]. Common challenges associated with biomedical data-sets, such as data processing speed, sparse labels, inconsistent staining, or laborious 3D imaging can be tackled with deep learning methods [12, 58, 66]. These new techniques promise to transform biomedical research and address fundamental questions from basic to translational science [12, 66]. The large amount of image data collected makes manual processing inefficient or even impossible. Machine learning methods allow for automated processing, de-noising, classification, visualization, data integration and comparison of large data-sets, and through statistical analysis they enable the interpretation of complex image data to extract new biological insights [12, 58]. By automatically learning complex patterns in images that may not be obvious to humans, machine learning algorithms can uncover hidden biological relationships and provide new insights from biological data [12, 24, 58]. Thereby, microscopic image analysis can be standardized and accelerated as machine learning tools are fast to apply and can deliver reliable analysis.

Deep learning (see section 1.5) has contributed to the advancement of microscopic image analysis in manifold applications.

Microscopic images can be classified (see section 1.3) with deep learning methods [24, 58], wthout relying on biological or hand-crafted features, deep learning models can identify different types of cells [61] or tissue [67]. With a classification method however only one class per image can be predicted, which complicates the analysis of large images where several different cells or structures might be present.

Object detection algorithms, such as the Mask R-CNN [68] (see section 1.6.4) have been developed to detect multiple different objects in one image. These deep learning models are costly to train as they require large annotated data-sets, but they deliver excellent performance once trained [69].

Biological structures can be segmented (see section 1.3.2) routinely with deep learning methods from fluorescent microscopic images to uncover their position, shape and morphology [70]. Recently, deep learning methods have become available delivering good performance without requiring the user to retrain the model [71, 72]. Morphological features, such as the volume, surface or eccentricity can be calculated from a segmentation [2], or features outputted by the deep learning model can be used for other tasks. However, the features are often not interpretable but might capture information that is missed using hand crafted features. The extraction of these cellular features is an important task, as they can provide information about cell type, cellular states or processes, which in turn can provide information about biological mechanisms or diseases.

In the field of drug discovery, extracted features have been used to reveal patterns indicating biological activity, to identify disease associated phenotypes, to understand disease mechanisms or to predict the activity, toxicity or mechanism of action of the drug [73].

In the application of flow cytometry, microscopic images of single cells can now be routinely analyzed and classified [11, 74]. Class frequency changes and morphological changes can be used to link morphological features to function, thereby machine learning techniques can revolutionize high-throughput experiments, for example to allow for antibody design [6]. Fluorescent markers can be predicted with machine learning algorithms from bright-field images in fixed tissue to uncover single cell morphology or state, without requiring laborious, time-consuming and phototoxic stains [28, 29, 75]. In an *in-silico* staining task, each pixel or voxel in a bright-field image is correlated to a predicted intensity that resembles a fluorescent staining.

Deep learning methods can also be used to accelerate microscopic image acquisition. Image restoration techniques can be used to under-sample the recording of image planes and use deep learning to predict missing information on a tissue level [76]. Furthermore, deep neural methods can virtually refocus a two-dimensional fluorescence image onto user-defined three-dimensional surfaces within the sample, obtaining 3D information [77]. However, deep learning algorithms typically require large annotated data-sets for training, assessment and testing of generalizability [12, 24, 58].

Unsupervised learning (see section 1.4.3) has recently emerged as a new field in deep learning, allowing researchers to train models without explicit labels. These methods rely on creating a supervised task on unlabelled data using implicit labels from the data-set to train a model [31]. It can be used to train models for image denoising [78], or for feature learning [6, 79], which can be used for gene expression prediction [80] or to identify and quantify microanatomical tissue structures in multiplexed images [79]. In the application of histopathology, unsupervised learning can be used to search large databases that contain gigapixel sized whole slide images to identify relevant regions in images [81]. Unsupervised learning has also been used in histopathology to identify sub-regions of high diagnostic value in whole slide images, to classify whole slides or to perform a clustering of image features [82]. Often, models trained in an unsupervised fashion are later refined on a smaller labeled data-set in a supervised fashion [74]. When applied successfully, deep learning algorithms can achieve higher accuracy in single cell classification tasks than trained experts [66, 83, 84] and outperform humans at data processing speed and prediction consistency [84–86].

Deep learning based image analysis can be extended with a new layer of information by using multi-omics approaches [87–89]. The combination with information from other modalities, such as genomics, epigenomics, transcriptomics, proteomics, and metabolomics, offers the opportunity to investigate relationships across multiple levels of cellular organization and to provide new insights into biological processes that cannot be inferred from a single modality [90, 91]. The integration of large, complex, multimodal data into a single model represents a considerable challenge [91].

First steps in the field have been made combining microscopy with genomics and transcriptomics but data integration remains a challenge. For survival outcome prediction, histology images and genomics have shown to complement each other leading to superior prognostic performance in cancer diagnosis and prognosis [92]. In combination with spatial transcriptomics, gene activity across different cells and intracellular structures can be studied. It has been shown that registration and analysis of transcriptome images and immunostaining images can reveal new biological insights [93].

1.8. Research questions

Deep learning can open new ways to extract information from microscopic images. This requires the development of reproducible and reliable methods that are easily applicable by researchers from different fields. Within the scope of this cumulative thesis, the following research questions are addressed:

I) How can a larger audience be enabled to apply state-of-the-art deep learning algorithms?

II) How can the effort of obtaining 3D morphological information of single cells from microscopy images be reduced?

III) How can single cells be monitored in 3D time-lapse bright-field movies?

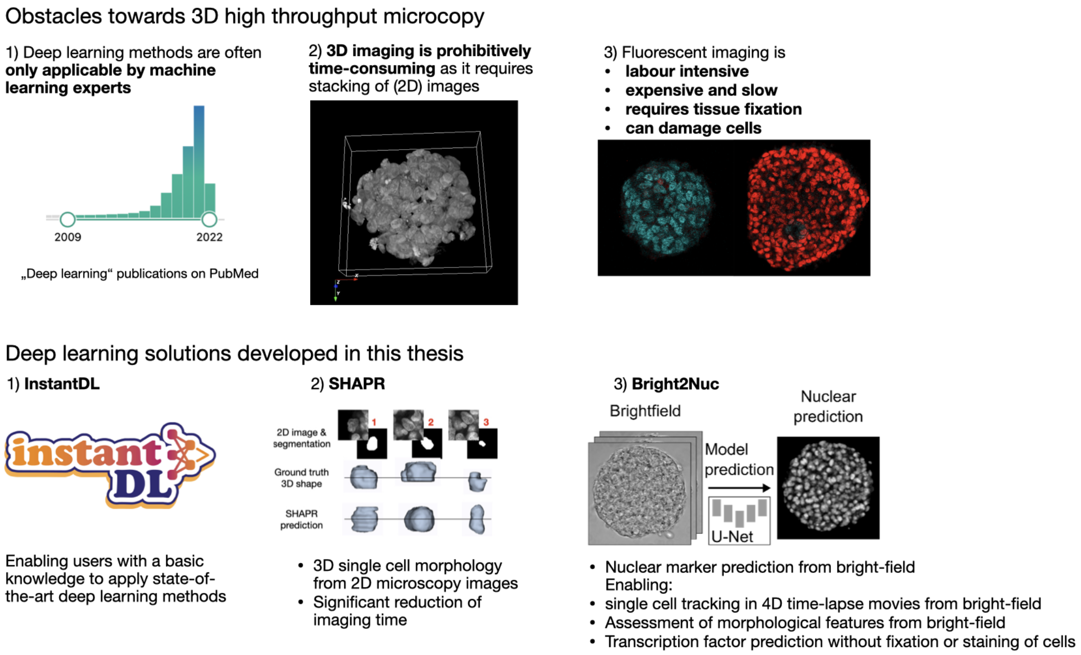
1.9. Summary of results

The series of papers I present in this thesis provide researchers with methods that can be used to simplify and accelerate the analysis of single cell image data-sets and thereby pave the way towards quantitative high-throughput microscopy (see figure 1.8). InstantDL [1] provides researchers with an easy-to-use, reliable framework, combining deep learning models for semantic segmentation, instance segmentation, pixel-wise regression and classification. This debugged and benchmarked framework eases the access for researchers with a basic computational background to state-of-the-art deep learning algorithms. The SHAPR [2] algorithm can be used to reconstruct the shape of 3D cells or nuclei from 2D images. This algorithm reduces the number of microscopic images needed to obtain 3D information and thereby minimizes the effort of microscopic imaging. SHAPR has been extended in a follow-up [3] with a topology-based loss function that captures contextual information and thereby further improves shape prediction. I further present Bright2Nuc [4], a deep learning model to predict a nuclear marker from 3D bright-field images, allowing for segmentation of nuclei from bright-field images, tracking nuclei in cell cultures imaged in live bright-field, and predicting cell states based on the expression of transcription factors from single cell morphology. Thereby researchers can omit tissue fixation and staining and still obtain a nuclear marker signal to assess cellular morphology. Furthermore, it allows for the tracking of single cells in 4D time-lapse movies.

Core publications as main author

- [1] in section 2.1: *InstantDL - An easy-to-use deep learning pipeline for image segmentation and classification*

  Most deep learning algorithms are developed to solve one specific problem; they typically require a considerable coding effort and a machine learning background for their application. This raises issues of reproducibility and applicability when trying to transfer the methods



Obstacles towards 3D high throughput microcopy
1) Deep learning methods are often only applicable by machine learning experts
2009
2022
„Deep learning“ publications on PubMed
2) 3D imaging is prohibitively time-consuming as it requires stacking of (2D) images
3) Fluorescent imaging is
• labour intensive
• expensive and slow
• requires tissue fixation
• can damage cells
Deep learning solutions developed in this thesis
1) InstantDL
instant DL
Enabling users with a basic knowledge to apply state-of-the-art deep learning methods
2) SHAPR
2D image & segmentation
Ground truth 3D shape
SHAPR prediction
1
2
3
• 3D single cell morphology from 2D microscopy images
• Significant reduction of imaging time
3) Bright2Nuc
Brightfield
Model prediction
U-Net
Nuclear prediction
• Nuclear marker prediction from bright-field Enabling:
• single cell tracking in 4D time-lapse movies from bright-field
• Assessment of morphological features from bright-field
• Transcription factor prediction without fixation or staining of cells

Figure 1.8: In this thesis, three obstacles for the application of high-throughput microscopy are tackled with deep learning methods. As the main author, I published three contributing articles, which have the potential to ease the application of deep learning methods in microscopic image analysis, enable researchers to obtain 3D morphological information of single cells from 2D microscopy images and provide tools to monitor single cells in 3D time-lapse movies.

developed to other data-sets. To ease application commonly used state-of-the-art deep learning algorithms for microscopic image analysis are bundled in InstantDL. InstantDL can be used for semantic segmentation, instance segmentation, pixel-wise regression and classification, providing automated and standardized workflows. The algorithm was extensively tested on different data-sets. InstantDL, despite not being tuned for a specific task, achieves competitive results on a variety of biomedical image data-sets that were used for benchmarking. It is in use at the Helmholtz Zentrum Munich and has been cited from researchers in several countries.

- [2] in section 2.2: *SHAPR - An AI approach to predict 3D cell shapes from 2D microscopic images*
  Imaging single cells in 3D is a time-consuming task as it requires stacking two-dimensional images. Our SHApe PRediction autoencoder (SHAPR) eases the trade-off between throughput and resolution in space and time for 3D microscopic imaging by predicting the 3D shapes of single cells from 2D microscope images. This deep learning algorithm combines a 2D encoder for feature extraction from 2D images with a 3D decoder to predict 3D shapes from latent space representations. The results demonstrate that SHAPR is able to predict 3D shapes of single red blood cells and shapes of cell nuclei imaged with a confocal microscope. The nuclei were counterstained human-induced pluripotent stem cells

(iPSCs) cultured in a spheroidal aggregate. The potential applications of SHAPR include accelerated drug screening with patient derived cell clusters.

- [3] in section 2.3: *Capturing Shape Information with Multi-Scale Topological Loss Terms for 3D Reconstruction*
  The challenging task of predicting 3D objects from 2D images is tackled by SHAPR [2]. To further improve this spatial reasoning task, contextual information of the shapes of the objects is critical. However, established loss functions only capture geometrical information, but not contextual information. Therefore, we complement an established loss function (e.g. the dice loss and binary cross entropy loss) with contextual information, by comparing multi-scale topological features, such as connected components, cycles and voids, between the ground truth and SHAPRs predictions. By adding this loss function to the SHAPR algorithm, we substantially improved reconstruction quality. Applications of this topological loss function span from image segmentation to image reconstruction and shape prediction with deep learning methods.

- [4] in section 2.4: *Label-free imaging of 3D pluripotent stem cell differentiation dynamics on chip*
  Organoids can replicate organ complexity or mimic functional aspects and thereby allow researchers to study organ development and disease. To unleash the full potential of automated high-throughput screening analysis of organoids with single cell resolution and thereby gain insight into organ development, self-organization and disease progression novel tools are required. These tools can extract information from bright-field images, making fixation and staining of tissue unnecessary and that can process large amounts of data, acquired during time-lapse experiments fast. We have developed Bright2Nuc, a deep learning method to predict nuclear staining from 3D bright-field images, reducing slow, expensive and phototoxic fluorescent stainings that can damage cells [4]. Furthermore, we can segment individual nuclei, quantify their morphological properties, track them over time and predict transcription factor expression. By applying our deep learning tools on organoids of human induced pluripotent stem cells differentiating towards definitive endoderm, we determine the number of nuclei with high accuracy, calculate nuclei velocities and predict lineage specific transcription factor expression. Our methods are available in an open source pipeline.

2. Methods: Summary of contributed articles

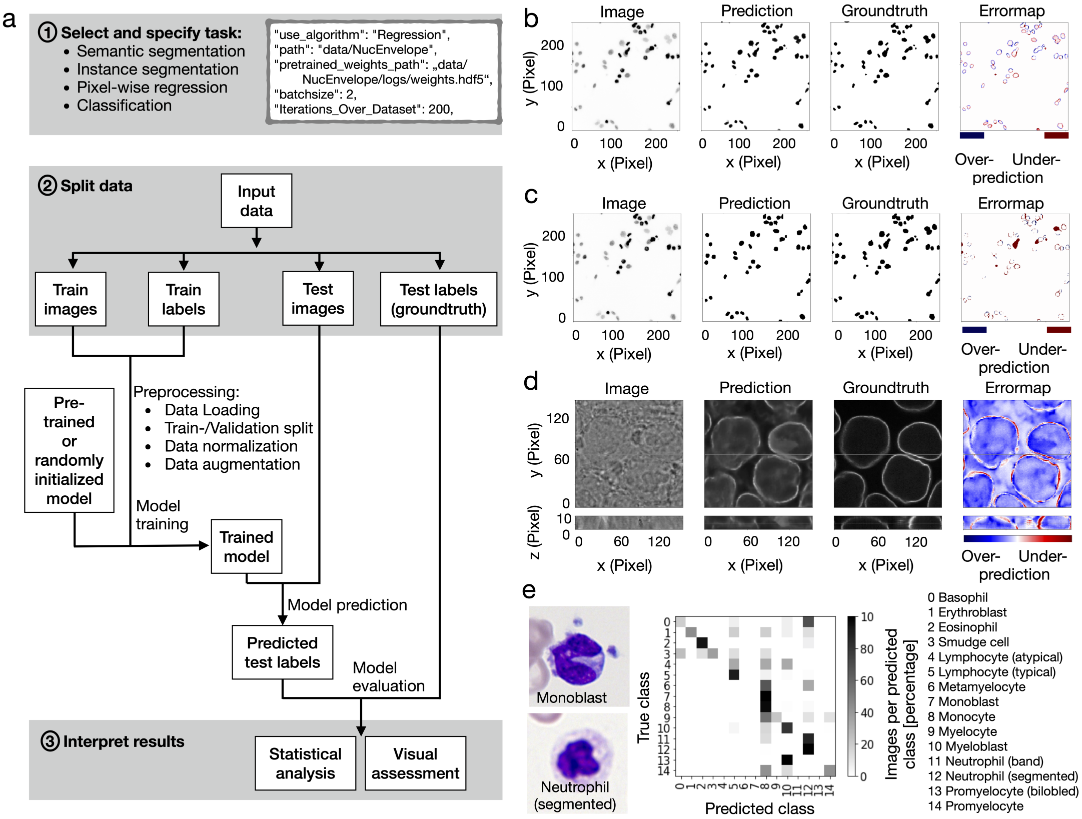
This chapter summarizes my contributed articles in the context of the methods described in section 1.

2.1. InstantDL — An easy-to-use deep learning pipeline for image segmentation and classification

Combining the architectures, methods and concepts described in section 1, I have built an easy-to-use deep learning pipeline, InstantDL [1].

Research problem
Published algorithms mostly solve only one specific problem as they are developed by one lab for a specific task. Plus they typically require a considerable coding effort and machine learning background for their usage. This raises issues of reproducibility and applicability for other labs, when trying to use the methods on other data-sets.

Approach
I have aimed at developing a deep learning pipeline for four common image processing tasks: semantic segmentation, instance segmentation, pixel-wise regression and classification. Therefore, I have used existing deep learning architectures and combined them in one pipeline: a U-Net (see section 1.6.6) for semantic segmentation and regression tasks, a Mask R-CNN for instance segmentation (see section 1.6.4), and a ResNet (see section 1.6.3) with 18 layers for image classification.
One of the main challenges was to unify the pre- and evaluation for these different methods and architectures, in order to make InstantDL easy and convenient to apply. To tackle this, I have implemented shared data loaders, data normalization, data augmentation and optimization procedures as described in section 1.6.1. To automate and thereby simplify the evaluation for the user, I have added commonly used metrics, which are automatically calculated after test set predictions. To assess the uncertainty of U-Net and ResNet predictions, I have added dropout layers (see section 1.5.2) to the model enabling the application of Monte Carlo dropout [45] as described in section 1.5.4. To run InstantDL, the user must only split the data-set in a train and test set by sorting the images in the corresponding folders, the validation set will automatically be split from the test set during dataloading. Thereafter, InstantDL can be run with default settings, meaning that the user must only provide the task name and data path. If the user wishes to adapt InstantDL, a maximum of twelve parameters can be set. These are: The task (i.e. semantic segmentation, instance segmentation, regression, or classification), a seed for reproducibility, path to the project directory containing the train and test files, pre-trained weights to initialize the model, batchsize, number of iterations over the data-set, data augmentations, loss function, number of classes, if the images should be resized during import, uncertainty calculation after training, and automated evaluation after training. With InstantDL I also provide ten dif-



```
"use_algorithm": "Regression",
"path": "data/NucEnvelope",
"pretrained_weights_path": „data/
        NucEnvelope/logs/weights.hdf5“,
"batchsize": 2,
"Iterations_Over_Dataset": 200,
```

Figure 2.1: InstantDL [1] provides researchers with an easy-to-use deep learning pipeline for biomedical image analysis. **a**, Indicated user actions are highlighted in gray in this flow diagram. (1) The user must choose the desired task (semantic segmentation, instance segmentation, pixel-wise regression, classification) and may choose from to twelve parameters in the configuration file, illustrated by a code snippet. (2) The user must split the data into train and test set by putting the data in the corresponding folders. By executing the python configuration file, the pipeline automatically loads the data, creates a validation set, normalizes and augments the data and then starts training using either a pre-trained or randomly initialized model. Once finished training, the model predicts test labels, and results can be interpreted by the user via statistical and visual assessment that is automatically performed by InstantDL. Example output for a **b**, 2D semantic segmentation using the U-Net, **c**, a 2D instance segmentation using the Mask R-CNN, a **d** 3D pixel-wise regression task using the U-Net, and **e**, a classification task using the ResNet.

ferent pretrained weights, which the user can use for transfer learning as described in section 1.5.3.

Validation on published data-sets

We have benchmarked InstantDL on seven publicly available data-sets (four of which come from data science challenges) and compared its performance to published results. If no test set was provided, we took 20% of the data to create our own test set. This was done on the highest level of abstraction, for example on patient level or tissue-slide level whenever possible, otherwise the data was randomly split. I used the same evaluation metrics as published in the respective publications (Jaccard index, AUC, Pearson correlation) to compare our results appropriately.

Results

InstantDL enables researchers with a basic computational background to apply debugged and benchmarked state-of-the-art deep learning algorithms to their own data with minimal effort. To make the pipeline robust, I have automated and standardized workflows and extensively tested it in different scenarios. With InstantDL, I achieved competitive results without any hyperparameter tuning. While I could not reach the challenge winners' performance (who typically use specifically tuned algorithms) of the respective challenges with our out-of-the-box approach, the output of InstantDL suffices for standard biomedical data analytics performed after the image processing step.

Conclusion

InstantDL simplifies the access to the advantages of deep learning algorithms for biomedical researchers with limited background in computer science. The only requirement is a solid understanding of the data (and how to appropriately split it into training and test set) and of the loss function that should be optimized during training the model. The pipeline is designed for maximal automation to make training and testing as convenient and easy as possible. However, some parameter settings depend on the data-set properties and therefore cannot be automated. After setting a maximum of twelve parameters, the pipeline can be run without further user interactions. We included state-of-the-art analysis metrics that are directly accessible out-of-the-box. Moreover, we incorporated uncertainty prediction to provide an additional level of interpretability. For customization of the pipeline to specific tasks, all code is easily accessible and well documented, see also A.1.

Individual contribution: Carsten Marr and Niklas Köhler had the idea to create an easy-to-use general deep learning pipeline that can be applied to common image analysis tasks. Together with Carsten Marr and Niklas Köhler, I developed the scope of InstantDL. I implemented the complete code: I wrote a U-Net for semantic segmentation and pixel-wise regression, added an open-source package of the Mask-RCNN for instance segmentation and an existing package containing a ResNet for image classification. I adapted these packages to create a smoothly and efficiently operating pipeline. I wrote the data loader, the data augmentation module, the evaluation pipeline, and the uncertainty estimation from scratch. I benchmarked InstantDL on

selected publicly available single cell image data-sets with the support of Niklas Kiermeyer. With Sayedali Shetab Boushehri, I created the python package and unit tests. Finally, I wrote the text and created the figures of the manuscript together with Sayedali Shetab Bousheri and Carsten Marr.
I am main the main author of this publication.

2.2. SHAPR predicts 3D cell shapes from 2D microscopic images

With the methods and concepts described in section 1, I have built a SHApe PRediction autoencoder (SHAPR) [2].

Research problem
Recording single cells in three dimensions (3D) as opposed to recording single cells in two dimensions (2D) is often desired, as it leads to an information gain in the obtained images. However, 3D imaging is prohibitively time-consuming as it requires stacking of two-dimensional (2D) images or tomographic imaging, limiting its application in high-throughput biomedical imaging. This raises the question of how throughput and resolution in space and time can optimally be traded off.

Approach
We aim to eliminate the trade-off between throughput and resolution in space and time for 3D microscopic imaging by predicting the 3D shapes of single cells from 2D microscope images. I have developed SHAPR, a deep learning algorithm that combines a 2D encoder for feature extraction from 2D images with a 3D decoder to predict 3D shapes from a latent space representation, resulting in an autoencoder-style model, as introduced in section 1.6.5. I have used a 2D encoder with three convolutional blocks and three downsampling steps to extract the information, and seven convolutional blocks with seven upsampling layers to obtain 3D shapes from the latent space representation.
To increase SHAPR's prediction accuracy, a discriminator model was trained to differentiate between true and predicted 3D cell shapes, similar to an an adversarial model, as described in section 1.4.2.

Data-sets
I have used two data-set to evaluate SHAPR. For proof of concept, I predicted cell shapes using a recently published library detailing 3D red blood cell shapes (n = 825 cells) [94, 95]. Each cell shape was reconstructed from 68 confocal images with a z-resolution of 0.3µm. The 2D image of the red blood cell is intersecting the 3D image at the center slice. In a second step, I challenged SHAPR to predict cell nuclei shapes from confocal z-stacks containing fluorescence counterstained nuclei from human-induced pluripotent stem cells (iPSCs) cultured in a spheroidal aggregate. To generate the ground truth data, 887 cell nuclei from six iPSC-derived aggregates were manually

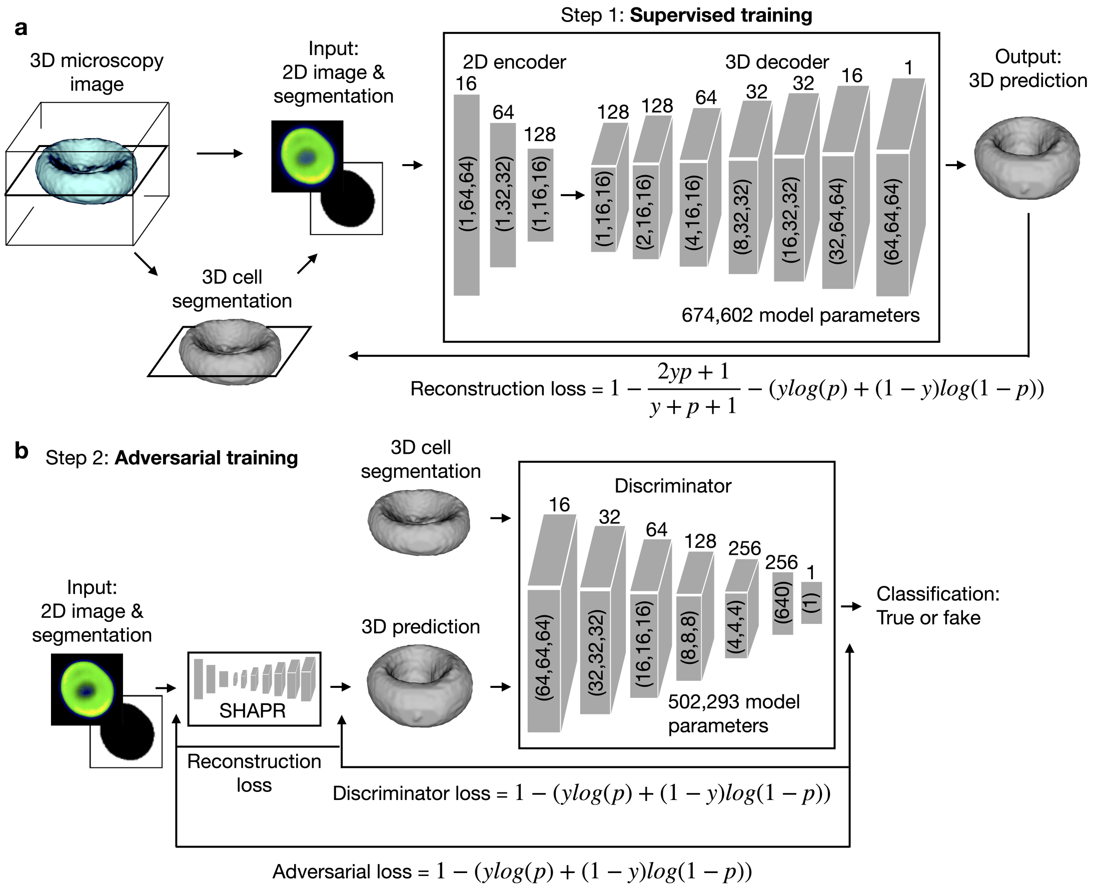


Figure 2.2: **a**, SHAPR [2] pairs a 2D encoder and a 3D decoder, which reconstructs a 3D shape from a 2D image embedding. To train SHAPR, we segment 3D microscopy images and combine each 2D segmentation with the microscopy image of the same slice to be used as an input for the encoder. We train SHAPR in two steps; first supervised, where the sum of the dice loss and the binary cross entropy loss between the 3D segmentations and SHAPR predictions is minimized. **b**, In a second step SHAPR is fine-tuned by adding a discriminator. The discriminator is trained to differentiate between the output of SHAPR and the ground truth segmentation and minimize the adversarial loss. It thereby challenges SHAPR to output realistic 3D objects.

segmented in 3D. SHAPR was provided with one 2D image slice taken at an aggregate depth of 22 µm in the cell culture and its corresponding segmentation as input, meaning that nuclei were cut at random heights, leading to a variety of segmented areas. I have made the datasets available via Zenodo [95].

Results

SHAPR is able to predict 3D shapes of single red blood cells and nuclei from 2D microscope images. Compared against two naïve stereological models, i.e. a cylindrical and an ellipsoid fit, the predictions of SHAPR significantly outperformed these models with respect to intersection over union, volume error, 2D surface area error, and surface roughness error. Using features of the predicted 3D red blood cells for a classification task into six classes, the F1 score was significantly improved by adding 3D prediction features of SHAPR to features extracted from the 2D images only.

Conclusion

I have shown that SHAPR can be used to reconstruct 3D cell shapes from 2D fluorescent microscopy images. From the predictions of SHAPR, relevant features can be extracted, as proven by a downstream classification task, which was significantly improved when adding 3D prediction features of SHAPR as opposed to using the features of the 2D images alone. Applying SHAPR to single nuclei, we showed that relevant nuclear characteristics are captured, as opposed to the two naïve models. Using SHAPR to predict 3D shapes from 2D images thus offers a simple way to reduce imaging time and data storage as compared to 3D microscopy, while retaining morphological details. Trained SHAPR models can be applied to predict single cell volume distributions and density, can screen cell cultures and identify outlier events. Future applications might also include feature based label-free transcription factor prediction or single cell classification for diagnostic purposes in computational pathology. SHAPR may also be extended to increase efficiency of biomedical imaging in multiple domains, including other biological structures, as for example organelles and proteins, see also A.2.

Individual contribution: Carsten Marr and I had the idea of predicting 3D cell morphology from 2D microscopy images. I was responsible for creating the computational methodology from scratch, including designing and implementing the code, the deep learning network, the data loader, the data augmentation and the evaluation scripts from scratch. With Carsten Marr, I designed the experiments and, together with Niklas Kiermeyer, I performed the hyperparameter optimization and conducted the experiments. With Mohammad Mirkazemi, I wrote the SHAPR python package to make the method publically available. Matthias Meier and Scott Atwell provided a data-set that Scott manually annotated for our evaluation. Finally, I created the text and figures of the manuscript with Scott Atwell, Ario Sadafi, Matthias Meier and Carsten Marr. I am main the main author of this publication.

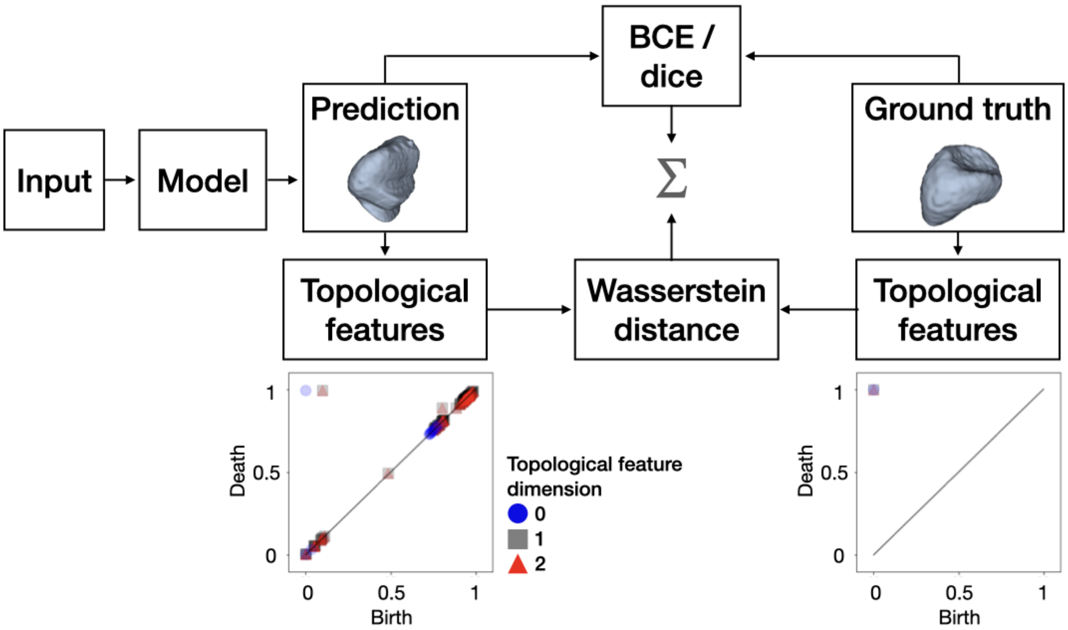


Figure 2.3: A novel topological loss function for capturing shape information on mutliple scales. For an object predicted by SHAPR and its corresponding 3D ground truth object, we calculate topological features with cubical persistent homology, thereby we obtain two persistence diagrams. Each point in a persistence diagram denotes the birth and death of a topological feature of some dimension of the given object in different scales. With our topological loss we compare these diagrams and combine this with existing loss terms like BCE and dice loss.

2.3. Capturing Shape Information with Multi-scale Topological Loss Terms for 3D Reconstruction

Using the methods described in section 1.5.5 and utilizing SHAPR (see section 2.2) as a use-case we have developed a novel topological loss function [3].

Research problem

Using SHAPR for reconstruction of 3D objects from 2D images is a spatial reasoning task. Contextual information about the overall shape of an object is not captured by established loss terms like the dice loss or BCE loss (see section 1.3.3).

Approach

We aimed to improve the reconstruction performance of SHAPR using a novel regularization technique that goes beyond geometrical loss terms, but captures contextual information. This contextual information is not caputres by established loss terms, such as the dice loss or BCE loss. Therefore we have developed a method to extract multi-scale topological features, compare these features with an optimal transport metric and use it as a loss function (see section 1.5.5). We jointly optimised the model using the BCE, dice loss and topological loss during training with $L = \frac{1}{2}L_{BCE} + \frac{1}{2}L_{Diceloss} + \lambda L_{topological}$, where $\lambda \in \mathbb{R}_{+}$. To accelerate the calculation of our topological loss term, we downsample each volume from 64×64×64 voxels to 16×16×16 voxels using trilinear interpolation. I found that this is sufficiently good enough to capture topological information, while resulting in only small empirical errors.

Data-sets

We have used the same two data-sets as for SHAPR (see section 2.2 and [2]).

Results

To evaluate the topological loss, I perform the same experiment twice: In a first round we used a joint BCE and dice loss (setting $\lambda = 0$), followed by adding the topological loss (setting $\lambda = 0.1$). Without the topological loss, I achieve results comparable to the original SHAPR publication [2], while minor deviations arise from stochasticity and implementation differences between PyTorch and Tensorflow. I jointly optimised the BCE, dice loss and topological loss during training, which lead to the best results in a hyperparameter search. We observe statistically significant performance gains in all but one of multiple metrics for both data-sets when adding the topological loss to the model [3].

Conclusion

I have shown that a topology-aware loss function can improve a spatial reasoning task, by employing multi-scale topological features. The loss term can be integrated into existing deep learning models, for reconstruction and segmentation tasks. It does not optimise classical segmentation or reconstruction metrics, but serves as an inductive bias for incorporating multi-scale topological features, capturing contextual information. I found that the topological loss did not improve performance evenly across different classes of the red blood cell data-set, but improves results the most on shapes that have complex morphological features, such as echinocytes or bi-concavely shaped cells (discocytes and stomatocytes) [3], see also A.3.

Individual contribution: Bastian Rieck and I discussed how we could improve the predictions of SHAPR with a content-aware loss function. Bastian had the idea to implement a topological loss. I translated SHAPR from tensorflow to pytorch, to enable the integration of the topological loss function that he created, and we both implemented the topological loss into the SHAPR framework. Together, we optimized hyperparameters and performed experiments. Bastian and I created the figures and, with the help of Carsten Marr, we wrote the manuscript.
I am main the main author of this publication.

2.4. Label-free imaging of 3D pluripotent stem cell differentiation dynamics on chip

Using the methods described in section 1 and based on InstantDL, described in section 2.1, I have build Bright2Nuc, enabling insights into cell differentiation.

Research problem *In-vitro* 3D stem cell cultures can mimic *in-vivo* environments and be used for culturing organ-specific tissue from stem cells. They can replicate organ complexity or express selected aspects and thereby allow researchers to study organ development and disease. To unleash the full potential of automated high-throughput screening analysis of 3D cell cultures, to gain insight into cell development, self-organization and disease progression, novel computational tools are needed. Currently, slow and expensive imaging in 3D using phototoxic fluorescent stains hampers its potential. Tissue requires to be fixed before staining, killing cells and making long term monitoring infeasible.

Approach

I have developed Bright2Nuc, a deep learning methods to predict nuclear staining in 3D from bright-field images, so tissue must not be fixed and stained to investigate cellular properties [4]. This *in-silico* staining of 3D bright-field images opens a range of downstream analysis like cell segmentation and analysis of morphological features or single cell tracking. I trained a U-Net (see section 1.6.6) on paired bright-field and fluorescently labeled images to predict a nuclear marker staining. To incorporate 3D information, I modified the U-Net with 3D convolutional layers and 3D downsampling and upsampling layers. In each downsampling and upsampling layer the x-y-dimension was changed, while the z-dimension stayed unchanged throughout the network. Thereby, neighboring slice information was added. Using this *in-silico* staining, I could segment individual nuclei using a re-trained method called Stardist [72, 96], allowing for feature based morphometry and cell status prediction. Applying the *in-silico* staining to live bright-field movies in 4D allowed me to track single nuclei in space and time using trackmate [97].

Data-sets

All the data used for this publication was imaged by Scott Atwell with a confocal microscope. I have made them available via Zenodo [98].

Results

We showcase our methods on data-sets containing aggregates of human induced pluripotent stem cells, imaged with a confocal microscope. In two experiments cell cultures have been imaged in live bright-field, and were later fixed and stained with a nuclear marker and with three transcription factor stainings. These transcription factor stainings indicating cell state, namely pluripotency or differentiated towards definitive endoderm. In two other experiments cell cultures have not been fixed and stained but bright-field movies were recorded. During one of these experiments a reporter cell line was used marking a sub-population of cells with a nuclear marker which was used as a benchmark. Our tools allow for:

- the calculation of the number of nuclei with high accuracy in fixed and live 3D bright-field images
- the calculation of nuclei velocities in live 4D bright-field movies

- the prediction of cell states from 3D bright-field images

We were able to confirm existing literature by showcasing that we measure an increase of nuclei movement during differentiation.

Conclusion

Our deep learning approaches allow for monitoring of single cells in cell cultures, reducing the need to fix and stain the tissue to investigate cellular morphology or cell states. Thereby I pave the way towards high-throughput applications microscopy based cell analysis and monitoring. By *in-silico* staining of bright-field images I can predict the nuclei position, shape and morphology and thereby gain insights into cell states and movement patterns, without the need for fixation and staining of tissue [4], see also A.4.

Individual contribution: Matthias Meier and Carsten Marr developed the idea to predict nuclear markers from the 3D bright-field images that Scott Atwell records in the laboratory. There was no applicable solution for this data available, therefore Matthias Meier, Scott Atwell, Carsten Marr and I created an outline for the project. I was responsible for adapting the InstantDL pipeline to the specific needs of predicting nuclear markers from 3D bright-field images. This enables tracking of individual nuclei over time. Moreover, it allows for segmentation the nuclei for assessing their morphology. From the morphology, we predicted the transcription factor expression of single nuclei. With Carsten Marr, Matthias Meier and Scott Atwell, I designed the computational experiments. I developed the Bright2Nuc python package to make the method publicly available. Matthias Meier and Scott Atwell provided the data-sets. Scott and I were responsible for creating the manuscript text and figures with the help of Carsten Marr and Matthias Meier.
Scott Atwell and I share main authorship of this publication.

3. Discussion

Deep learning models have become drivers of innovation in biomedical microscopy data analysis with numerous applications, such as denoising, illumination correction, image restoration, stain prediction, segmentation, feature extraction and morphology analysis. Being able to outperform non-machine learning methods and human experts concerning data processing speed, deep learning methods are well suited to handle the increasing amount of microscopy images and larger 3D data-sets. Another advantage of deep learning methods is their prediction consistency as, once trained, most of the deep learning models are deterministic. They provide consistent predictions as opposed to a human who's predictions might vary.

3.1. Summary

In this thesis, I developed deep learning methods for 3D microscopy image analysis. The resulting tools extended the impact that researchers can generate from their microscopy image data. The presented methods are easy-to-apply, benchmarked, reproducible, reviewed, well-documented, and were published open source. They can be used for image segmentation and classification, 3D shape prediction from 2D images as well as *in-silico* staining.

InstantDL [1] enables users to apply state-of-the-art deep learning methods in an easy-to-use framework for recurring image analysis tasks, such as image segmentation, pixel-wise regression or image classification (see section 2.1).

To accelerate microscopic imaging, I developed SHAPR [2], a deep learning model that predicts the 3D shape of single cells and nuclei from 2D microscopic images. This enables researchers to retrieve 3D morphological information of single cells, while only imaging in 2D (see section 2.2). By replacing 3D imaging with 2D imaging, which is multiple times faster, SHAPR can substantially decrease imaging time and allow for high-throughput microscopy.

The latest developments at the intersection between mathematics and computer science enabled the fast extraction of topological features from images [49, 99, 100]. We extended SHAPR with a topological loss function [3], which considers contextual information, based on the extraction of multi-scale topological features using a cubical complex. These topological features are compared using Wasserstein distance, a metric from optimal transport. The developed loss function is model agnostic and can be added to any neural network (see section 2.3).

To access currently untapped information contained in bright-field images, such as the single cell volume, I developed Bright2Nuc, a deep learning model to predict a nuclear marker from 3D bright-field images [4]. Bright2Nuc allows researchers to segment in-silico stained nuclei and to assess morphological information in bright-field images (see section 2.4). By expanding the *in-silico* staining method to live 4D bright-field movies (3 spatial dimensions and time), the method allows quantification of single cell dynamics. Furthermore, I was able to predict cell states that indicate the differentiation status with a feature-based approach by combining morphological features extracted from bright-field images and the nucleus predictions from Bright2Nuc.

Taken together, these methods enable faster image analysis: InstantDL eases the access to state-of-the-art deep learning methods, SHAPR accelerates 3D microscopic imaging, and Bright2Nuc reduces the number of fluorescent stains required to assess the position, shape, morphology and dynamics of a single cell. All presented methods increase the amount of information that researchers can automatically extract from microscopic images. This thesis benefits the research community as I ensured reproducible results by providing well-documented and reviewed open-access python packages, by meeting high software standards, and by benchmarking my work on several data-sets.

The developed methods can pave the way towards microscopy-based, non-invasive, high-throughput single cell monitoring and image analysis enabled by deep learning methods. The methods presented in this thesis enable researchers to apply state-of-the-art deep learning methods to their own data, to reduce microscopic imaging time, the use of fluorescent stains and tissue fixation, and enable monitoring of cells in high-throughput time-lapse experiments, only based on bright-field movies. Thereby, the impact that researchers can generate from microscopy experiments can be increased.

3.2. Outlook

Developments in deep learning

New deep learning algorithms are constantly developed, with ever increasing predictive performance and new applications, as indicated by the exponentially increasing number of publications in the field. These new deep learning methods will significantly change 3D microscopic imaging towards high-throughput cell monitoring in space and time. This development will thereby help to uncover previously undetectable information, such as phenotypes, from bright-field images.

Label scarcity is an ever occurring issue when training deep learning models on biomedical data. These labels are especially difficult to obtain in the biological and medical domain, as an expert is often required to perform annotations [101]. Self supervised learning can help utilizing unlabelled data-sets in numerous applications, as it can be used to train deep learning models without explicit labels (see section 1.4.4). Self supervised models can then either be directly applied on specific tasks, such as to discover tissue architecture in multiplex imaging [79], or in a transfer learning approach, where a model is trained in a self supervised fashion on a data-set without annotations and then the model weights are refined in another supervised training step [6]. In future, I expect to see the field of self supervised deep learning methods expanding, whereby unlabeled data is used effectively. Presumably, most models will be initialized with weights obtained from a self supervised tasks, and shared between applications, as transfer learning improves model performance and decreases training time, thus saving valuable resources.

Approaches that use weakly labelled data-sets are emerging, meaning that insufficient annotations are available, for example. if only a fraction of the data-set is labelled or only higher level labels are available. Examples are large whole slide images that are too large to be processed by a computer at once and only contain labels for the respective whole slide image but not for sub-regions of the image. Multiple instance learning methods are developed for these weakly labeled data-set, they intelligently extract features from image sub-regions and combine them so that supervised training is enabled [102].

Recently, diffusion models have gained increasing attention due to their predictive power [103, 104], outperforming other generative models. They have also found application on biological and medical tasks, for anomality detection [105] and for image segmentation [106]. We applied a diffusion model to predict 3D single cell shapes from 2D images, yielding superior results [10]. I expect these models to find wide application in microscopic image analysis, for example, in domain transfer or to impute missing data.

Graph neural networks are another emerging method that has the potential to find a wide range of use cases in microscopic image analysis. Single cells are not isolated objects, but must be understood in the context of their environment. Accordingly, graph neural networks are able to efficiently describe relationships between tissue regions and cells [107]. They have already been successfully applied in histopathology to retrieve regions of interest in whole slide images [108] and for image classification [107, 109, 110]. Graph deep learning is to detect contextual pathological features within a complex tumor microenvironment for prognosis prediction of patients with tumours and to identify prognostic biomarkers [111]. Furthermore, graph neural networks can leverage spatial protein profiles in tissue specimens to model tumour microenvironments to detect contextual pathological features. Thereby, not only prognostic biomarkers can be identified, but they can also capture distinctive cellular interactions associated with differential clinical outcomes [112]. In the application of transcription factor prediction, I expect a graph neural network to outperform our feature based regression approach [4] as graph neural networks can relate a single nucleus to its context and thereby increase predictive performance.

Transformer models [113] have started to outperform traditional deep learning methods in medical image segmentation [114] and classification [115]. Transformer models are originally sequence-to-sequence models with the obvious application in natural language processing. They transform a sequence, such as the words in a sentence, into another sequence. This task can be reformulated so that transformers are applicable on images [116]. These architectures provide exciting opportunities for a new generation of deep learning methods.

Applicability and reproducibility

The field of microscopic imaging lacks behind in terms of standardization and reproducibility of techniques as compared to genomic sciences. In the field of genomic sciences, next-generation sequencing techniques in combination with robust platforms for profiling gene expression, high-quality shared databases and analysis tools have contributed to a major success, including hundreds of genome-wide association studies [7, 8, 12]. Automatic image analysis tools are often developed to address one specific problem in an individual project or lab, leading to many highly specialized solutions. Due to the high specificity of the tools, the provided solutions are often not easily applicable by other researchers and therefore do not scale [7, 117, 118].

Tackling this issue, general purpose deep learning frameworks, such as OpenML [119], offer an ecosystem of data-sets and machine learning models. However, online models require that the users upload their data-set and make it publicly available, which is often not possible in the biomedical domain because of strict data security restrictions. ImJoy [120] is an online tool with that users can select plugins for processing their data, which is easy and convenient, yet not locally executable. The tools nucleAIzer [121] and yeastspotter [122] are easy-to-use web tools but provide only cell and nuclei segmentation using drag-and-drop with pre-trained machine learning models.

Data-sets

Large standardized and public data-sets, such as ImageNet [62] in the natural image domain, are likewise required in the biomedical domain to test, benchmark and compare the developed methods. This is partly tackled by machine learning challenges, often accompanying machine learning conferences, where participants can test and compare methods [123, 124]. However, these challenges are often performed on a limited data-set in terms of number and diversity of images, ensuring neither the generalization of developed methods to other data-sets nor their applicability by other researchers. In recent years, data-sets have been used to define a challenge that motivated researches to develop solutions for a specific problem, thereby the challenge organizers have set a direction for the field, for example in nucleus segmentation [125]. Creating challenges that solve meaningful problems of a field is one powerful way of moving a research field forward. In future, I expect to see large, curated multi-omics data-sets that are continuously updated in terms of number of data points, annotations and modalities to create a standard for researches to test and benchmark their tools. Larger data-sets might also require the application of multiple methods, such as modality specific pre-processing and feature extraction, in combination with data integration and evaluation tools.

For the future, I also envision easier-to-apply deep learning methods, such as InstantDL, making them usable for a large audience, for example by their integration into user-friendly interfaces such as ImageJ [126] and Napari [127], as exemplified by TrackMate [128]. This approach will, furthermore, simplify the successive application of deep learning methods, for example performing a noise reduction followed by a staining prediction and a cell segmentation. Currently, these are separate steps for researchers, requiring different methods that all have to be successively applied. Solving these challenges can lead to faster, more reliable and more reproducible microscopy image analysis that enables high-throughput imaging.

Multi-omics integration

In combination with multi-scale and multi-omics approaches [87–89], high-throughput microscopy approaches, such as SHAPR and Bright2Nuc, can amplify the information researchers are able to obtain in biomedical experiments. Multi-omics, including image-based cell profiling, can add a new layer of information to existing approaches, that can complement existing analysis. The field still has a lot of untapped potential, with a few challenges remaining, such as data integration. Recently a multi-modal approach displayed the fields potential, by integrating medical imaging, histopathologic and genomic features to predict immunotherapy response [129]. In future multi-omics approaches can help to uncover numerous biological phenomena, help to identify relevant genes via functional genomics, to study cell responses to treatments, and to detect new biomarkers [130].

Contribution to translational research

Large-scale image-based screenings can accelerate several fields, such as drug screening and drug discovery, in-vitro disease modeling, personalized medicine and regenerative therapies [17, 130]. In the application of drug screening, image-based cell profiling can be used to extract information from biological images to uncover relevant patterns, reveal unexpected biological activity and, thereby, contribute to the identification of disease-associated phenotypes, disease mechanisms and prediction of drug activity [130]. Moreover, disease diagnosis [131, 132] and prediction of patient outcomes in cancer research and in histopathology [133–135] can profit from light microscopy in combination with deep learning for image analysis by accelerating diagnosis and increasing prediction performance. Personalized medicine can benefit from deep learning methods applied on light microscopy images, for measuring patient responses after treatment with various potential therapies [130].

For a clinical application of deep learning methods, many technical, regulatory and ethical challenges remain. The numerous obstacles for implementation of machine learning methods in daily clinical practice include issues with transparency, the inherent bias in the data they are fed with and potential security [136]. As a first step, the machine learning model needs to be thoroughly evaluated and regulatory authorities need to approve the method, e.g., the U.S. Food and Drug Administration (FDA) or the European Medicines Agency (EMA) [136, 137]. Joint efforts of interdisciplinary teams and strong communication between data scientists and clinicians can span the gap between machine learning research and its adoption in the clinic [137]. All these issues can be addressed through adequately planned validation studies, leading to credible clinical implementations of deep learning in the clinics [138] in close collaboration with regulatory authorities and clinical partners.

The mentioned novel deep learning techniques promise to transform biomedical research and address fundamental questions from basic to translational science, offering an exciting outlook for biomedical researchers.

Bibliography

[1] D. J. E. Waibel, S. Shetab Boushehri, and C. Marr. “InstantDL: an easy-to-use deep learning pipeline for image segmentation and classification”. *BMC Bioinformatics* 22.1 (Mar. 2021), p. 103.

[2] D. J. Waibel et al. “SHAPR predicts 3D cell shapes from 2D microscopic images”. *iScience* 25.11 (2022), p. 105298. ISSN: 2589-0042. DOI: https://doi.org/10.1016/j.isci.2022.105298. URL: https://www.sciencedirect.com/science/article/pii/S258900422201570X.

[3] D. J. E. Waibel et al. “Capturing Shape Information with Multi-scale Topological Loss Terms for 3D Reconstruction”. *Medical Image Computing and Computer Assisted Intervention – MICCAI 2022* (2022), pp. 150–159. eprint: https://doi.org/10.1007/978-3-031-16440-8_15.

[4] S. Atwell et al. “Label-free imaging of 3D pluripotent stem cell differentiation dynamics on chip”. *bioRxiv* (2022). eprint: 10.1101/2022.08.30.505835.

[5] D. J. E. Waibel et al. “In-Silico Staining from Bright-Field and Fluorescent Images Using Deep Learning”. *Artificial Neural Networks and Machine Learning – ICANN 2019: Image Processing.* Ed. by I. V. Tetko et al. Cham: Springer International Publishing, 2019, pp. 184–186. ISBN: 978-3-030-30508-6.

[6] S. Shetab Boushehri et al. “Systematic Comparison of Incomplete-Supervision Approaches for Biomedical Image Classification”. *Artificial Neural Networks and Machine Learning – ICANN 2022.* Ed. by E. Pimenidis et al. Cham: Springer International Publishing, 2022, pp. 355–365. ISBN: 978-3-031-15919-0.

[7] S. J. Wagner et al. “Make deep learning algorithms in computational pathology more reproducible and reusable”. *Nature Medicine* (2022), pp. 1–3.

[8] S. J. Wagner et al. “Built to last? Reproducibility and Reusability of Deep Learning Algorithms in Computational Pathology”. *medRxiv* (2022).

[9] B. Mairhörmann et al. “Deep learning-based magnetic resonance imaging lung segmentation and volumetric marker extraction in preterm infants”. Aug. 2021. eprint: 10.1101/2022.05.15.22275108.

[10] D. J. Waibel et al. “A diffusion model predicts 3d shapes from 2d microscopy images”. *arXiv preprint arXiv:2208.14125* (2022).

[11] S. S. Boushehri et al. “PXPermute: Unveiling staining importance in multichannel fluorescence microscopy”. *bioRxiv* (2023), pp. 2023–05.

[12] N. Bagheri et al. “The new era of quantitative cell imaging—challenges and opportunities”. *Mol. Cell* 82.2 (Jan. 2022), pp. 241–247.

[13] S. Skylaki, O. Hilsenbeck, and T. Schroeder. “Challenges in long-term imaging and quantification of single-cell dynamics”. en. *Nat. Biotechnol.* 34.11 (Nov. 2016), pp. 1137–1144.

[14] M. Li et al. “Multimodality reporter gene imaging: Construction strategies and application”. en. *Theranostics* 8.11 (Apr. 2018), pp. 2954–2973.

[15] D. J. Stephens and V. J. Allan. “Light microscopy techniques for live cell imaging”. en. *Science* 300.5616 (Apr. 2003), pp. 82–86.

[16] G. Kayser and K. Kayser. “Quantitative pathology in virtual microscopy: history, applications, perspectives”. en. *Acta Histochem.* 115.6 (July 2013), pp. 527–532.

[17] H. Chen et al. “The rise of deep learning in drug discovery”. en. *Drug Discov. Today* 23.6 (June 2018), pp. 1241–1250.

[18] J. W. Lichtman and J.-A. Conchello. “Fluorescence microscopy”. en. *Nat. Methods* 2.12 (Dec. 2005), pp. 910–919.

[19] J. Valli et al. “Seeing beyond the limit: A guide to choosing the right super-resolution microscopy technique”. en. *J. Biol. Chem.* 297.1 (July 2021), p. 100791.

[20] J. Icha et al. “Phototoxicity in live fluorescence microscopy, and how to avoid it”. en. *Bioessays* 39.8 (Aug. 2017).

[21] J. Jonkman et al. “Tutorial: guidance for quantitative confocal microscopy”. en. *Nat. Protoc.* 15.5 (May 2020), pp. 1585–1611.

[22] Y. Avior, I. Sagi, and N. Benvenisty. “Pluripotent stem cells in disease modelling and drug discovery”. en. *Nat. Rev. Mol. Cell Biol.* 17.3 (Mar. 2016), pp. 170–182.

[23] T. Mitchell et al. “Machine Learning”. *Annu. Rev. Comput. Sci.* 4.1 (June 1990), pp. 417–433.

[24] K. P. Murphy. *Probabilistic Machine Learning: An Introduction.* en. MIT Press, Feb. 2022.

[25] L. R. Dice. “Measures of the Amount of Ecologic Association Between Species”. *Ecology* 26.3 (1945), pp. 297–302.

[26] A. Reinke et al. “Metrics Reloaded - A new recommendation framework for biomedical image analysis validation”. Apr. 2022.

[27] Fuyong Xing et al. “Deep Learning in Microscopy Image Analysis: A Survey”. en. *IEEE Trans Neural Netw Learn Syst* 29.10 (Oct. 2018), pp. 4550–4568.

[28] C. Ounkomol et al. “Label-free prediction of three-dimensional fluorescence images from transmitted-light microscopy”. en. *Nat. Methods* 15.11 (Nov. 2018), pp. 917–920.

[29] E. M. Christiansen et al. “In Silico Labeling: Predicting Fluorescent Labels in Unlabeled Images”. en. *Cell* 173.3 (Apr. 2018), 792–803.e19.

[30] A. Kurakin, I. J. Goodfellow, and S. Bengio. “Adversarial machine learning at scale. 2017”. *URL https://arxiv. org/abs/1611. 01236* (2017).

[31] A. Esteva et al. “Deep learning-enabled medical computer vision”. en. *NPJ Digit Med* 4.1 (Jan. 2021), p. 5.

[32] L. Breiman. “Random Forests”. *Mach. Learn.* 45.1 (Oct. 2001), pp. 5–32.

[33] M. Li, M. Soltanolkotabi, and S. Oymak. “Gradient Descent with Early Stopping is Provably Robust to Label Noise for Overparameterized Neural Networks”. *Proceedings of the Twenty Third International Conference on Artificial Intelligence and Statistics.* Ed. by S. Chiappa and R. Calandra. Vol. 108. Proceedings of Machine Learning Research. PMLR, 2020, pp. 4313–4324.

[34] D. P. Kingma and J. Ba. “Adam: A Method for Stochastic Optimization”. *arXiv:1412. 6980 [cs]* (Dec. 2014).

[35] K. He et al. “Mask r-cnn”. *Proceedings of the IEEE international conference on computer vision.* 2017, pp. 2961–2969.

[36] M. Abadi et al. “TensorFlow: Large-Scale Machine Learning on Heterogeneous Distributed Systems” (Mar. 2016). arXiv: 1603.04467 `[cs.DC]`.

[37] A. Paszke et al. “PyTorch: An Imperative Style, High-Performance Deep Learning Library”. *Advances in Neural Information Processing Systems.* Ed. by H. Wallach et al. Vol. 32. Curran Associates, Inc., 2019.

[38] Baldi and Sadowski. “Understanding dropout”. *Adv. Neural Inf. Process. Syst.* ().

[39] Y. Gal and Z. Ghahramani. “Dropout as a Bayesian Approximation: Representing Model Uncertainty in Deep Learning” (June 2015). arXiv: 1506.02142 `[stat.ML]`.

[40] C. Garbin, X. Zhu, and O. Marques. “Dropout vs. batch normalization: an empirical study of their impact to deep learning”. *Multimed. Tools Appl.* 79.19 (May 2020), pp. 12777–12815.

[41] C. Shorten and T. M. Khoshgoftaar. “A survey on Image Data Augmentation for Deep Learning”. *Journal of Big Data* 6.1 (July 2019), p. 60.

[42] L. Taylor and G. Nitschke. “Improving Deep Learning with Generic Data Augmentation”. *2018 IEEE Symposium Series on Computational Intelligence (SSCI).* Nov. 2018, pp. 1542–1547.

[43] A. Mikołajczyk and M. Grochowski. “Data augmentation for improving deep learning in image classification problem”. *2018 International Interdisciplinary PhD Workshop (IIPhDW).* May 2018, pp. 117–122.

[44] Y. Wei et al. “Conditional single-view shape generation for multi-view stereo reconstruction”. *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition.* 2019, pp. 9651–9660.

[45] Y. Gal and Z. Ghahramani. “Dropout as a Bayesian Approximation: Representing Model Uncertainty in Deep Learning”. en. *International Conference on Machine Learning.* June 2016, pp. 1050–1059.

[46] C. Tan et al. “A Survey on Deep Transfer Learning”. *Artificial Neural Networks and Machine Learning – ICANN 2018.* Springer International Publishing, 2018, pp. 270–279.

[47] M. Pandey et al. “The transformational role of GPU computing and deep learning in drug discovery”. en. *Nature Machine Intelligence* 4.3 (Mar. 2022), pp. 211–221.

[48] H. B. Griffiths. *Basic topology, by M. A. Armstrong. Pp 250. £19·90. 1983. ISBN 3-540-90839-0 (Springer).* 1985.

[49] J. Clough et al. “A Topological Loss Function for Deep-Learning based Image Segmentation using Persistent Homology”. en. *IEEE Trans. Pattern Anal. Mach. Intell.* PP (Sept. 2020).

[50] H. Lebesgue. “Sur les correspondances entre les points de deux espaces”. *Fund. Math.* 2.1 (1921), pp. 256–285.

[51] A. R. Pears. *Dimension theory of general spaces.* Cambridge university press, 1975.

[52] Edelsbrunner et al. “Topological Persistence and Simplification”. *Discrete & Computational Geometry* 28.4 (2002), pp. 511–533.

[53] H. Fan, H. Su, and L. Guibas. “A Point Set Generation Network for 3D Object Reconstruction from a Single Image”. *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* (2017).

[54] C. Hofer et al. “Graph filtration learning”. *International Conference on Machine Learning.* PMLR. 2020, pp. 4314–4323.

[55] M. Moor et al. “Topological Autoencoders”. *Proceedings of the 37th International Conference on Machine Learning.* Ed. by H. D. Iii and A. Singh. Vol. 119. Proceedings of Machine Learning Research. PMLR, 2020, pp. 7045–7054.

[56] A. Poulenard, P. Skraba, and M. Ovsjanikov. “Topological function optimization for continuous shape matching”. *Computer Graphics Forum.* Vol. 37. 5. Wiley Online Library. 2018, pp. 13–25.

[57] M. Carriere et al. “Optimizing persistent homology based functions”. *International conference on machine learning.* PMLR. 2021, pp. 1294–1303.

[58] E. Moen et al. “Deep learning for cellular image analysis”. en. *Nat. Methods* 16.12 (Dec. 2019), pp. 1233–1246.

[59] C. Angermueller et al. “Deep learning for computational biology”. en. *Mol. Syst. Biol.* 12.7 (July 2016), p. 878.

[60] A. Cutler, D. R. Cutler, and J. R. Stevens. “Random Forests”. *Ensemble Machine Learning: Methods and Applications.* Ed. by C. Zhang and Y. Ma. Boston, MA: Springer US, 2012, pp. 157–175.

[61] C. Matek, S. Schwarz, K. Spiekermann, et al. “Human-level recognition of blast cells in acute myeloid leukaemia with convolutional neural networks”. *Nature Machine* (2019).

[62] J. Deng et al. “Imagenet: A large-scale hierarchical image database”. *2009 IEEE conference on computer vision and pattern recognition.* Ieee. 2009, pp. 248–255.

[63] D. Bank, N. Koenigstein, and R. Giryes. "Autoencoders" (Mar. 2020). arXiv: 2003.05991 [cs.LG].

[64] O. Ronneberger, P. Fischer, and T. Brox. "UNet Convolutional Networks for Biomedical Image Segmentation". *arXiv:1505. 04597 [cs]* (May 2015).

[65] N. Siddique et al. "U-Net and Its Variants for Medical Image Segmentation: A Review of Theory and Applications". *IEEE Access* 9 (2021), pp. 82031–82057.

[66] A. Esteva et al. "Dermatologist-level classification of skin cancer with deep neural networks". en. *Nature* 542.7639 (Feb. 2017), pp. 115–118.

[67] Z. Liu et al. "A survey on applications of deep learning in microscopy image analysis". *Computers in Biology and Medicine* 134 (2021), p. 104523.

[68] K. He et al. "Mask r-cnn". *Proceedings of the IEEE international conference on computer vision.* 2017, pp. 2961–2969.

[69] J. C. Caicedo et al. "Nucleus segmentation across imaging experiments: the 2018 Data Science Bowl". *Nature methods* 16.12 (2019), pp. 1247–1253.

[70] J. C. Caicedo et al. "Evaluation of Deep Learning Strategies for Nucleus Segmentation in Fluorescence Images". en. *Cytometry A* 95.9 (Sept. 2019), pp. 952–965.

[71] C. Stringer et al. "Cellpose: a generalist algorithm for cellular segmentation". en. *Nat. Methods* 18.1 (Jan. 2021), pp. 100–106.

[72] U. Schmidt et al. "Cell Detection with Star-Convex Polygons". *Medical Image Computing and Computer Assisted Intervention – MICCAI 2018.* Springer International Publishing, 2018, pp. 265–273.

[73] S. N. Chandrasekaran et al. "Image-based profiling for drug discovery: due for a machine-learning upgrade?" *Nature Reviews Drug Discovery* 20.2 (2021), pp. 145–159.

[74] S. S. Boushehri et al. "scifAI: Explainable machine learning for profiling the immunological synapse and functional characterization of therapeutic antibodies". en. Oct. 2022.

[75] Y. Rivenson et al. *PhaseStain: the digital staining of label-free quantitative phase microscopy images using deep learning.* 2019.

[76] M. Weigert et al. "Content-aware image restoration: pushing the limits of fluorescence microscopy". en. *Nat. Methods* 15.12 (Dec. 2018), pp. 1090–1097.

[77] Y. Wu et al. *Three-dimensional virtual refocusing of fluorescence microscopy images using deep learning.* 2019.

[78] W. Khademi et al. "Self-supervised poisson-gaussian denoising". *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision.* 2021, pp. 2131–2139.

[79] J. Kim et al. "Unsupervised discovery of tissue architecture in multiplexed imaging". *Nature Methods* (2022), pp. 1–9.

[80] J. T. Ash et al. "Joint analysis of expression levels and histological images identifies genes associated with tissue morphology". *Nature communications* 12.1 (2021), pp. 1–12.

[81] C. Chen et al. "Fast and scalable search of whole-slide images via self-supervised deep learning". *Nature Biomedical Engineering* (2022), pp. 1–15.

[82] M. Y. Lu et al. "Data-efficient and weakly supervised computational pathology on whole-slide images". *Nature biomedical engineering* 5.6 (2021), pp. 555–570.

[83] C. Matek et al. "Human-level recognition of blast cells in acute myeloid leukemia with convolutional neural networks". *bioRxiv* (2019).

[84] P. Tschandl et al. "Comparison of the accuracy of human readers versus machine-learning algorithms for pigmented skin lesion classification: an open, web-based, international, diagnostic study". *Lancet Oncol.* 20.7 (2019), pp. 938–947.

[85] X. Liu et al. "A comparison of deep learning performance against health-care professionals in detecting diseases from medical imaging: a systematic review and meta-analysis". *The lancet digital health* 1.6 (2019), e271–e297.

[86] V. Gulshan et al. "Performance of a Deep-Learning Algorithm vs Manual Grading for Detecting Diabetic Retinopathy in India". en. *JAMA Ophthalmol.* 137.9 (Sept. 2019), pp. 987–993.

[87] G. Palla et al. "Squidpy: a scalable framework for spatial omics analysis". en. *Nat. Methods* 19.2 (Feb. 2022), pp. 171–178.

[88] J.-K. Hériché, S. Alexander, and J. Ellenberg. "Integrating Imaging and Omics: Computational Methods and Challenges". *Annu. Rev. Biomed. Data Sci.* 2.1 (July 2019), pp. 175–197.

[89] L. Moses and L. Pachter. "Museum of spatial transcriptomics". en. *Nat. Methods* 19.5 (May 2022), pp. 534–546.

[90] M. Kang, E. Ko, and T. B. Mersha. "A roadmap for multi-omics data integration using deep learning". *Briefings in Bioinformatics* 23.1 (2022), bbab454.

[91] Z. Miao et al. "Multi-omics integration in the age of million single-cell data". *Nature Reviews Nephrology* 17.11 (2021), pp. 710–724.

[92] R. J. Chen et al. "Pathomic fusion: an integrated framework for fusing histopathology and genomic features for cancer diagnosis and prognosis". *IEEE Transactions on Medical Imaging* (2020).

[93] Y. Qiang et al. "A Framework for Registration of Multi-modal Spatial Transcriptomics Data". *International Conference on Pattern Recognition and Artificial Intelligence.* Springer. 2022, pp. 134–145.

[94] G. Simionato et al. "Red blood cell phenotyping from 3D confocal images using artificial neural networks". en. *PLoS Comput. Biol.* 17.5 (May 2021), e1008934.

[95] D. Waibel et al. *Datasets for 3D shape reconstruction from 2D microscopy images.* Zenodo, Sept. 2022. DOI: 10.5281/zenodo.7031924. URL: https://doi.org/10.5281/zenodo.7031924.

[96] M. Weigert et al. "Star-convex polyhedra for 3d object detection and segmentation in microscopy". *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision.* 2020, pp. 3666–3673.

[97] J.-Y. Tinevez et al. "TrackMate: An open and extensible platform for single-particle tracking". en. *Methods* 115 (Feb. 2017), pp. 80–90.

[98] S. Atwell et al. "Label-free imaging of 3D pluripotent stem cell differentiation dynamics on chip". *bioRxiv* (2022), pp. 2022–08.

[99] S. Shit et al. "clDice – a Novel Topology-Preserving Loss Function for Tubular Structure Segmentation" (Mar. 2020). arXiv: 2003.07311 `[cs.CV]`.

[100] X. Hu et al. "Topology-preserving deep image segmentation". *Adv. Neural Inf. Process. Syst.* 32 (2019).

[101] S. Shurrab and R. Duwairi. "Self-supervised learning methods and applications in medical imaging analysis: A survey". *PeerJ Computer Science* 8 (2022), e1045.

[102] S. Banerji and S. Mitra. "Deep learning in histopathology: A review". en. *Wiley Interdiscip. Rev. Data Min. Knowl. Discov.* 12.1 (Jan. 2022).

[103] Dhariwal and Nichol. "Diffusion models beat gans on image synthesis". *Adv. Neural Inf. Process. Syst.* ().

[104] A. Q. Nichol and P. Dhariwal. "Improved Denoising Diffusion Probabilistic Models". *Proceedings of the 38th International Conference on Machine Learning.* Ed. by M. Meila and T. Zhang. Vol. 139. Proceedings of Machine Learning Research. PMLR, 2021, pp. 8162–8171.

[105] J. Wolleb et al. "Diffusion Models for Medical Anomaly Detection". *arXiv preprint arXiv:2203.04306* (2022).

[106] J. Wolleb et al. "Diffusion Models for Implicit Image Segmentation Ensembles". *arXiv preprint arXiv:2112.03145* (2021).

[107] D. Ahmedt-Aristizabal et al. "A survey on graph-based deep learning for computational histopathology". *Computerized Medical Imaging and Graphics* (2021), p. 102027.

[108] Y. Ozen et al. "Self-supervised learning with graph neural networks for region of interest retrieval in histopathology". *2020 25th International Conference on Pattern Recognition (ICPR).* IEEE. 2021, pp. 6329–6334.

[109] J. Levy et al. "Topological feature extraction and visualization of whole slide images using graph neural networks". *BIOCOMPUTING 2021: Proceedings of the Pacific Symposium.* World Scientific. 2020, pp. 285–296.

[110] P. Pati et al. "Hierarchical graph representations in digital pathology". *Medical image analysis* 75 (2022), p. 102264.

[111] "Graph deep learning detects contextual prognostic biomarkers from whole-slide images". en. *Nat Biomed Eng* (Aug. 2022).

[112] Z. Wu et al. *Graph deep learning for the characterization of tumour microenvironments from spatial protein profiles in tissue specimens.* 2022.

[113] Vaswani, Shazeer, Parmar, et al. "Attention is all you need". *Adv. Neural Inf. Process. Syst.* ().

[114] D. Karimi, H. Dou, and A. Gholipour. "Medical Image Segmentation Using Transformer Networks". en. *IEEE Access* 10 (Mar. 2022), pp. 29322–29332.

[115] D. Reisenbüchler et al. "Local Attention Graph-based Transformer for Multi-target Genetic Alteration Prediction". *arXiv preprint arXiv:2205.06672* (2022).

[116] A. Dosovitskiy et al. "An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale" (Oct. 2020). arXiv: 2010.11929 `[cs.CV]`.

[117] F. Renard et al. "Variability and reproducibility in deep learning for medical image segmentation". en. *Sci. Rep.* 10.1 (Aug. 2020), p. 13724.

[118] M. B. A. McDermott et al. *Reproducibility in machine learning for health research: Still a ways to go.* 2021.

[119] J. Vanschoren. *OpenML.* https://www.openml.org. Accessed: 2020-1-10.

[120] W. Ouyang et al. "ImJoy: an open-source computational platform for the deep learning era". en. *Nat. Methods* 16.12 (Dec. 2019), pp. 1199–1200.

[121] R. Hollandi et al. "nucleAIzer: A Parameter-free Deep Learning Framework for Nucleus Segmentation Using Image Style Transfer". *Cell Systems* (May 2020).

[122] A. X. Lu et al. "YeastSpotter: accurate and parameter-free web segmentation for microscopy images of yeast cells". *Bioinformatics* (2019).

[123] J. C. Caicedo et al. "Nucleus segmentation across imaging experiments: the 2018 Data Science Bowl". en. *Nat. Methods* 16.12 (Dec. 2019), pp. 1247–1253.

[124] N. Kumar et al. "A Multi-Organ Nucleus Segmentation Challenge". en. *IEEE Trans. Med. Imaging* 39.5 (May 2020), pp. 1380–1391.

[125] J. C. Caicedo et al. "Nucleus segmentation across imaging experiments: the 2018 Data Science Bowl". en. *Nat. Methods* 16.12 (Dec. 2019), pp. 1247–1253.

[126] C. A. Schneider, W. S. Rasband, and K. W. Eliceiri. "NIH Image to ImageJ: 25 years of image analysis". en. *Nat. Methods* 9.7 (July 2012), pp. 671–675.

[127] C.-L. Chiu, N. Clack, et al. "napari: a Python Multi-Dimensional Image Viewer Platform for the Research Community". *Microscopy and Microanalysis* 28.S1 (2022), pp. 1576–1577.

[128] D. Ershov et al. "Bringing TrackMate into the era of machine-learning and deep-learning". en. Sept. 2021. eprint: 2021.09.03.458852.

[129] R. S. Vanguri et al. “Multimodal integration of radiology, pathology and genomics for prediction of response to PD-(L) 1 blockade in patients with non-small cell lung cancer”. *Nature cancer* 3.10 (2022), pp. 1151–1164.

[130] S. N. Chandrasekaran et al. “Image-based profiling for drug discovery: due for a machine-learning upgrade?” en. *Nat. Rev. Drug Discov.* 20.2 (Feb. 2021), pp. 145–159.

[131] M. Doan and A. E. Carpenter. “Leveraging machine vision in cell-based diagnostics to do more with less”. en. *Nat. Mater.* 18.5 (May 2019), pp. 414–418.

[132] J. D. Fauw et al. *Clinically applicable deep learning for diagnosis and referral in retinal disease.* 2018.

[133] Y. Zhu et al. *An image informatics pipeline for imaging mass cytometry to characterize the immune landscape in pre- and on-treatment immune therapy and its application in recurrent platinium-resistant epithelial ovarian cancer.* 2019.

[134] P. Mobadersany et al. “Predicting cancer outcomes from histology and genomics using convolutional networks”. en. *Proc. Natl. Acad. Sci. U. S. A.* 115.13 (Mar. 2018), E2970–E2979.

[135] H. W. Jackson et al. *The single-cell pathology landscape of breast cancer.* 2020.

[136] S. Benjamens, P. Dhunnoo, and B. Meskó. “The state of artificial intelligence-based FDA-approved medical devices and algorithms: an online database”. *NPJ digital medicine* 3.1 (2020), pp. 1–8.

[137] A. Couckuyt et al. “Challenges in translational machine learning”. *Human Genetics* (2022), pp. 1–16.

[138] X. Zhang et al. “Prospective Clinical Research of Radiomics and Deep Learning in Oncology: A Translational Review”. *Critical Reviews in Oncology/Hematology* (2022), p. 103823.

A. Appendix

A.1. InstantDL - An easy-to-use deep learning pipeline for image segmentation and classification

Dominik J. E. Waibel, Sayedali Shetab Boushehri, and Carsten Marr

This is a published version of the article in BMC Bioinformatics following peer review. The article is open-access thus the published version is inserted here.

Waibel *et al. BMC Bioinformatics* (2021) 22:103
https://doi.org/10.1186/s12859-021-04037-3

BMC Bioinformatics

SOFTWARE Open Access

InstantDL: an easy-to-use deep learning pipeline for image segmentation and classification

Dominik Jens Elias Waibel[1,2], Sayedali Shetab Boushehri[1,2,3] and Carsten Marr[1*]

*Correspondence: carsten.marr@helmholtz-muenchen.de
[1] Institute of Computational Biology, Helmholtz Zentrum München - German Research Center for Environmental Health, Neuherberg, Germany
Full list of author information is available at the end of the article

Abstract

Background: Deep learning contributes to uncovering molecular and cellular processes with highly performant algorithms. Convolutional neural networks have become the state-of-the-art tool to provide accurate and fast image data processing. However, published algorithms mostly solve only one specific problem and they typically require a considerable coding effort and machine learning background for their application.

Results: We have thus developed InstantDL, a deep learning pipeline for four common image processing tasks: semantic segmentation, instance segmentation, pixel-wise regression and classification. InstantDL enables researchers with a basic computational background to apply debugged and benchmarked state-of-the-art deep learning algorithms to their own data with minimal effort. To make the pipeline robust, we have automated and standardized workflows and extensively tested it in different scenarios. Moreover, it allows assessing the uncertainty of predictions. We have benchmarked InstantDL on seven publicly available datasets achieving competitive performance without any parameter tuning. For customization of the pipeline to specific tasks, all code is easily accessible and well documented.

Conclusions: With InstantDL, we hope to empower biomedical researchers to conduct reproducible image processing with a convenient and easy-to-use pipeline.

Background

Deep learning has revolutionised image processing [1]. For specific biomedical image analysis tasks such as cell segmentation [2, 3], cell classification [4–6] or in-silico staining [7, 8], deep learning algorithms now achieve higher accuracy than trained experts [6, 9, 10] and outperform humans at data processing speed and prediction consistency [11, 12]. However, machine learning algorithms are mostly developed to solve one specific problem. Moreover, applying them often requires a strong computer science and machine learning background.

We here provide InstantDL, a pipeline that automates pre- and post-processing for biomedical deep learning applications. InstantDL bundles commonly used deep learning algorithms in one easy-to-use framework. We did not change the bundled algorithms or tune them for a specific problem. Still, InstantDL achieves competitive results on different biomedical image datasets. It can be used for semantic segmentation (i.e. the classification of each pixel into a particular class), instance segmentation (i.e. the detection and classification of objects), pixel-wise regression (i.e. for in-silico staining) and image classification (i.e. to discriminate cancerous from healthy cells). InstantDL is an easy-to-use python package that can be executed by setting only two variables: the task and the data directory. For advanced users, twelve parameters can be set to adapt InstantDL to the specific task (see Methods). Moreover, different examples are provided in the documentation, which the user can follow. We provide 10 pre-trained models for transfer learning, since biomedical datasets are often sparsely annotated, as manual annotation is laborious and costly.

InstantDL is benchmarked on seven publicly available (see "Results" section) datasets: nuclei detection in divergent images [13], multi-organ nuclei segmentation [14, 15], lung segmentation from CT scans [16], in-silico prediction of a mitochondrial and a nuclear envelope staining [7], cell classification in digitized blood smears [17, 18], and cancer classification on histopathology slides [19]. Without any hyperparameter tuning, we achieve competitive results.

InstantDL serves as a centralized repository and provides a straight-forward wrapper to execute image computing tasks. It is designed for users interested in applying deep learning to their own data. A basic understanding of programming suffices. In contrast to easy-to-use browser based segmentation tools [3, 20], it is not limited to one specific task and ensures data privacy. The code is open source and well documented for those who want to customize the pipeline to their needs. By providing a debugged, tested, and benchmarked pipeline we help reduce errors during code development and adaptation, and contribute to reproducible application of deep learning methods.

Implementation

InstantDL offers the four most common tasks in medical image processing: Semantic segmentation, instance segmentation, pixel-wise regression, and classification [21, 22]. In the following we describe the algorithms implemented in InstantDL to address these tasks and how they can be applied within the pipeline.

Semantic segmentation

One of the standard approaches for detecting image patterns is semantic segmentation [23]. For each pixel in the input image a class label is predicted by the algorithm. The U-Net [24] is a commonly used architecture for semantic segmentation with numerous applications in biomedicine [25, 26]. It consists of a symmetric contractive path to capture context and an expansive path to capture fine localizations [24]. In InstantDL we made minor changes to the architecture from [24]: we (i) use padded convolutions to receive the same output and input dimensions and (ii) have implemented dropout layers in the encoder. These do not affect the performance of the network, but enable uncertainty quantification. The U-Net outputs continuous values between 0 and 1 for each pixel, which can be interpreted as probabilities to belong to a given class. InstantDL allows for two classes (background vs. foreground) and thresholds the output using Otsu' method [27, 28].

Instance segmentation

Instance segmentation is used to detect objects (instances) within an image [23, 28]. We implemented the Mask-RCNN [29] in InstantDL for this task. It first detects a bounding box for each object in the image and then performs a segmentation in each bounding box. Our Mask-RCNN is based on a ResNet50 from Abdullah's [30] implementation. The Mask-RCNN requires instance level ground truth: For each image in the training set, a set of labels has to be created, each containing a segmentation mask of one single instance. An algorithm to create instance level ground truth from binary semantic segmentation ground truth is provided as a jupyter-notebook with the pipeline.

Pixel-wise regression

Tasks where no pixel-wise class labels but a continuous pixel value is desired (such as in-silico staining, [7]) are called pixel-wise regression. InstantDL uses the same U-Net implementation as for semantic segmentation. The only difference is that the U-Net output is not interpreted as probabilities to belong to one or another class, but is regarded as a regression. We thus use continuous labels for training and the mean-squared-error as regressive loss as proposed previously [7].

Image classification

Here, the task is to classify each image into one of a specific number of given classes. For this task a residual network [31] is implemented. These architectures are widely used for biomedical image classification [12, 32]. Residual networks use residual blocks, a reformulation of layers as learning residual functions, which enable the use of many layers, while ensuring convergence [31]. We use a slightly modified ResNet50 with 50 layers in InstantDL, where we have added dropout layers to enable uncertainty estimation.

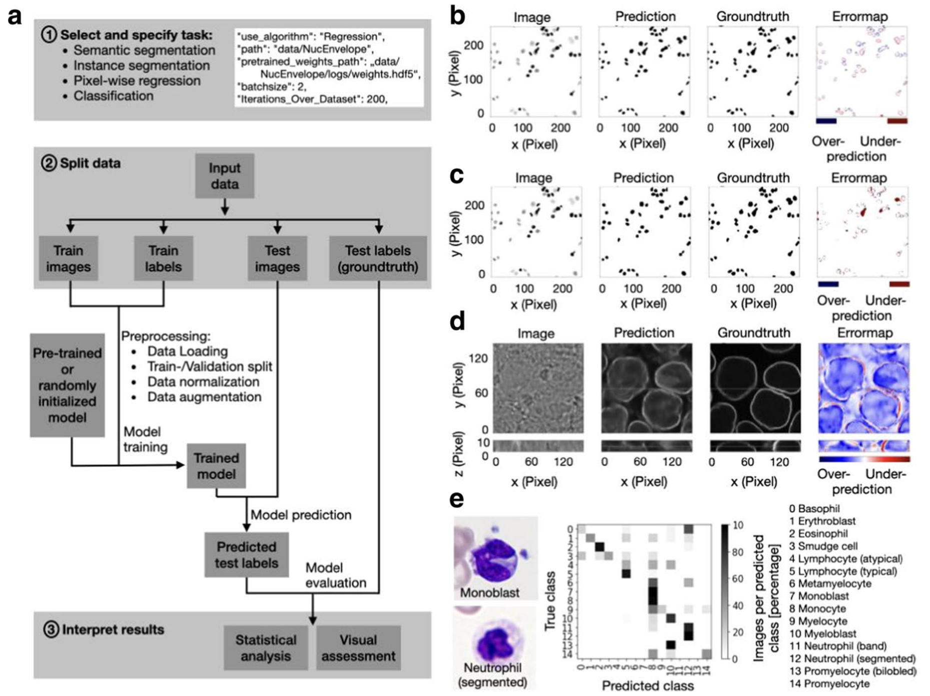
The InstantDL pipeline

Data preparation

Data has to be manually split in a train and a test set (see Fig. 1a) according to the user's hypothesis: For one dataset a random split might be suitable [33], while for others a split on patient or tissue slide level is methodically appropriate [6]. InstantDL can process stacked images, enabling the prediction from multiple channels. Input and ground truth images must have the same filename including the file ending (Table 1).

Pipeline settings

After data preparation the user can specify tasks and parameters in the configuration file (see Fig. 1a) or run InstantDL with default settings only providing the task and data path. A maximum of twelve parameters can be set. These are: The task (i.e. Semantic segmentation, instance segmentation, regression, or classification), the seed for reproducibility, the path to the project directory containing the train and test files, the pre-trained weights for model initialization, the batchsize, the number of iterations over the dataset (i.e. epochs), the data augmentations, the loss function, the



```
"use_algorithm": "Regression",
"path": "data/NucEnvelope",
"pretrained_weights_path": „data/
      NucEnvelope/logs/weights.hdf5",
"batchsize": 2,
"Iterations_Over_Dataset": 200,
```

Fig. 1 InstantDL provides an easy to use pipeline for the analysis of biomedical images. **a** Flow diagram of the pipeline with indicated user action highlighted in gray. (1) One out of four tasks (semantic segmentation, instance segmentation, pixel-wise regression, classification) is selected by the user. Up to twelve parameters can be set in the configuration file to adapt the pipeline to the task. A code snippet illustrates task selection and six of the twelve parameter settings in the configuration file: selected task ("use_algorithm"), path to folder ("path"), if pre-trained weights should be used the path to these ("pretrained_weights_path") should be set, batch size ("batchsize") and epochs chosen ("Iterations_Over_Dataset"). (2) Input data is split into train and test sets. The user specifies these by putting the data in the corresponding folders. After executing the python configuration file the pipeline will automatically load the data from the train folder, create a 20 percent validation split, normalize and augment the data (see Methods for details). Training is initiated with either a pre-trained or randomly initialized model. After training, the model predicts test labels: segmentation masks, pixel values or labels for the images in the test set according to the chosen task. (3) Results can be interpreted by the user via statistical and visual assessment of the predicted outcome by comparing it to the ground truth in the test set. **b** Example output for a 2D semantic segmentation task: Cell nuclei in a brightfield image (left) are segmented with InstantDL (Prediction) using the U-Net, and compared to the original annotation (Groundtruth). The Errormap indicates over- and under-predicted pixels. The image is part of the 2018 Kaggle nuclei segmentation challenge dataset [13]. **c** Example output for a 2D instance segmentation task (same image as in **b**): A binary mask is predicted for each object in the image using InstantDLs Mask-RCNN algorithm and compared to the groundtruth. **d** Example output for a 3D pixel-wise regression task using a U-Net. From stacks of bright-field images (Image) [7] the pipeline predicts a nuclear envelope (Prediction) that resembles the true staining (Groundtruth). The first row shows the x–y-plane, the bottom row the x–z plane of the 3D volume. **e** Example output for a classification task of benign and leukemic blood cells in blood smears from 200 individuals [17]. We show two exemplary microscopy images (left) of two white blood cell classes, a monoblast and a neutrophil. The white blood cell type is predicted with a ResNet50. The confusion matrix (middle) shows that most of the 15 classes can be well predicted, in accordance to Matek et al. [6]

number of classes, if the images should be resized during import, if the uncertainty should be calculated after training, and if the model should be automatically evaluated after training. Within the pipeline one.json file serves as a config file.

Table 1 Overview on the image processing tasks implemented in InstantDL, required input, label, and output format

	Semantic segmentation	Instance segmentation	Pixel-wise regression	Classification
Input image	2D & 3D	2D	2D & 3D	2D
Labels	Binary images	Images with float pixel values	Images with float pixel values	Labels in a *.csv* file
Output	Binary images	Binary masks	Images with float pixel values	Labels in a *.csv* file
Architecture	U-Net	Mask RCNN	U-Net	ResNet50

After setting these parameters, the user executes the configuration file which starts the training with InstantDL on the desired task using the training data with the pre-trained weights and the chosen configurations.

Transfer learning

Pre-trained weights can be used to initialize a training process, a practice called transfer learning. The choice of weights can have a huge influence on the performance of the algorithm. Depending on the dataset and problem, transfer learning from natural images such as ImageNet to the medical domain can or cannot be the preferred solution [34]. With InstantDL we provide 10 pre-trained weight sets: four pre-trained weights from 2D nuclei segmentation (two from semantic and instance segmentation of the nuclear detection challenge dataset [13] and the nuclei in microscopy images of multiple organs segmentation, respectively [14, 15]), two pre-trained weights from 2D lung segmentation [16] (from semantic and instance segmentation), two from 3D in-silico staining [7] (from predicting mitochondria and the nuclear envelope from brightfield images), and one from the classification of white blood cells [17, 18] and metastatic cancer [19], respectively. Moreover, ImageNet weights can be loaded. InstantDL will load the weights given in the configuration file. It automatically selects weights for layers which fit the desired model and reports them in the logfile. As it is not clear which pre-trained weight will lead to a performance increase on the desired task we recommend to use weights from a task with similar characteristics to the dataset at hand in terms of structure and color. Expert users can choose to only train desired layers in the model file.

Data augmentation

Data augmentation is a method commonly used in machine learning to artificially increase the variance in the training dataset and thereby train the network to generalize better [35]. We implemented spatial (horizontal and vertical flip, zoom and rotation) and color (contrast, brightness, poisson noise, feature scaling, standard-mean normalization, resampling, gamma shift) augmentations. The user can choose the desired augmentations, which are then randomly applied online, while importing the input images. The user can choose to visualize the augmentations used for training to ensure sensible inputs.

Model training

InstantDL reads data from the corresponding folders and prepares for training and testing. This includes the following steps: (i) Model initialization with pre-trained weights, if selected. (ii) Import of data and normalization, split of validation data from the data contained in the train data folder, shuffle of training data, batch creation and online data augmentation. (iii) Training of the model using the Adam optimizer [36] for the given number of epochs using early stopping, which can be monitored live using tensorboard and automated saving of the best model. (iv) Prediction of labels from the test dataset. (v) For semantic segmentation, pixel-wise regression and classification uncertainty can be calculated after training.

Model evaluation

The trained model is evaluated on the unseen test images and labels (i.e. the groundtruth). For semantic segmentation, instance segmentation and pixel-wise regression, the network predictions are saved as image stacks to ease evaluation of large datasets with limited CPU capabilities. This also allows an initial manual, qualitative evaluation and quality control. In the second step the predictions can be quantitatively evaluated. For that, accuracy, mean relative and absolute error, pixel-wise Pearson correlation coefficient and Jaccard index over all pixels of the test set are calculated. Boxplots are generated to visualize quantitative model performance. The standard quantitative evaluation output plots (i) the input images side-by-side to the corresponding labels and predictions and (ii) an error map between the labels and predictions to visualize training performance (see example evaluation Fig. 1b–d). For classification the predicted labels in the test set are compared to the true labels and multiple error scores (Jaccard index, mean absolute error, mean squared error, area under curve) are calculated. A confusion matrix and a receiver operating characteristic (ROC) curve are automatically visualized (Fig. 1e). All evaluation steps are implemented in the pipeline and can be set to be executed after testing. Additionally, post-processing (i.e. statistical analysis and visual assessment) is accessible in jupyter notebooks for customization, which are provided with InstantDL.

Uncertainty quantification

Neural networks predictions can be unreliable when the input sample is outside of the training distribution, the image is corrupted or if the model fails. Uncertainty estimation can measure prediction robustness, adding a new level of insights and interpretability of results [37, 38]. Bayesian inference can be approximated in deep Gaussian processes by Monte Carlo dropout [37]. We have implemented Monte Carlo dropout for semantic segmentation, pixel-wise regression and classification in the pipeline. During the inference phase, 20 different models are created using Monte Carlo dropout and model uncertainty is calculated on the test set. For pixel-wise regression and semantic segmentation, the pipeline saves an uncertainty map. From this, pixel uncertainty can be plotted and saved to the project folder or average pixel uncertainty for an image can be calculated (Fig. 2a, b). For classification InstantDL adds

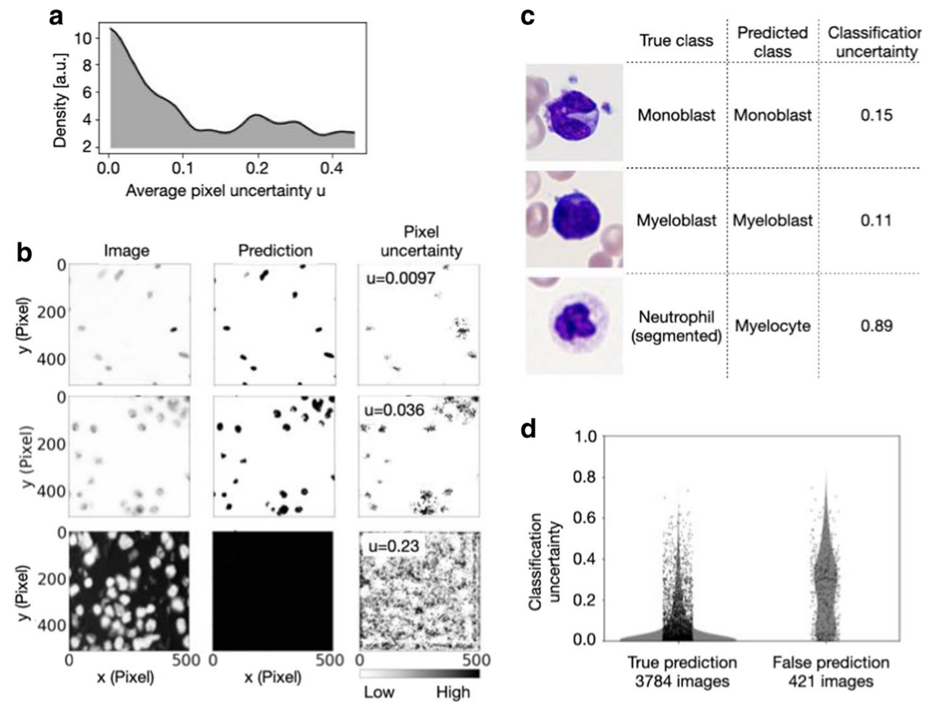


Fig. 2 Uncertainty of semantic segmentation, pixel-wise regression and classification can be estimated with InstantDL. **a** The distribution of average pixel uncertainty u for 162 images of a semantic segmentation task [13]. The distribution is approximately bi-modal. **b** Three exemplary semantic segmentations from the data visualized in **a** with InstantDLs pixel uncertainty estimation. Correct predictions correspond to a low average pixel uncertainty u (top and middle row), while a high average pixel uncertainty indicates failed segmentations (bottom row). Regions with ambiguous predictions are indicated by high pixel uncertainty (right column). **c** For the prediction of white blood cells classes [6], classification uncertainty indicates incorrect predictions. **d** The distributions of classification uncertainty for correct and false predictions differ significantly (*p* value < 0.001, Mann–Whitney rank test)

the uncertainty score to the results file where a score close to zero indicates certain predictions, and score close to 1 indicates high uncertainty (Fig. 2c, d).

Implementation of InstantDL

InstantDL is implemented in Python and leverages Tensorflow and Keras [39], which provide an excellent framework for our pipeline due to the modular, composable and user friendly design. The pipeline can be run locally ensuring data protection, on a cluster or with Google-Colab [40], which has successfully been used for deep learning projects [41] making it usable for those with limited computational resources. We provide the pipeline as one package in a Docker image [42] to simplify installation and application.

Hardware requirements

InstantDL can run on a CPU with or without GPU acceleration, locally or on a server. For users with limited hardware InstantDL can also be run using Google Colab. The notebook to execute InstantDL in Google Colab is provided with the package. On an Nvidia GeForce RTX2070, the multi-organ segmentation dataset requires 8 min to train with semantic segmentation and 14 min with instance segmentation. Semantic

(See figure on next page.)

Fig. 3 InstantDL achieves competitive performance on published datasets and computer vision challenges without hyperparameter tuning. **a** InstantDLs instance segmentation achieves competitive results on the nuclear detection challenge dataset [13], which contains a variety of experimental conditions, cell types, magnifications, and imaging modalities. We show one exemplary image from the dataset and the corresponding prediction using InstantDL's instance segmentation. The winner of the challenge achieved a Jaccard index of 0.63 (solid line), while the median participant achieved 0.42 (dotted line). InstandDLs instance segmentation achieved a median Jaccard index of 0.60 without hyperparameter tuning. We estimate the Jaccard index distribution by bootstrapping, sampling 100 times half of the test set. Boxes indicate the median and the 25/75%ile of the distribution, whiskers indicate the 1.5 interquartile range. **b** For the challenge of segmenting nuclei in microscopy images of multiple organs with hematoxylin and eosin staining [14, 15], the winner achieved a Jaccard index of 0.69 (solid line) and the median participant 0.63 (dotted line). InstantDL using instance segmentation reached a Jaccard index of 0.29, and 0.57 using semantic segmentation. **c** Evaluation of instance segmentation of lung CT images from the Vessel-12 dataset [16]. The winner of the challenge reached an area under the receiver operating characteristic curve (AUC) of 0.99, while the median participant reached 0.94. InstantDL reached an AUC of 0.90 with semantic segmentation, and 0.94 with instance segmentation. **d** InstantDL's pixel-wise regression performs similarly well as the published approach ([7] for in-silico staining of bright-field images in three dimensions, but with a higher variability. We achieved a median pearson correlation of 0.85 for nuclear envelope staining and 0.78 for mitochondria staining. **e** For classification of leukemic blast cell images vs. benign white blood cell images [17, 18], InstantDL achieved an AUC of 0.99, while Matek et al. report 0.99. **f** Classification of metastatic cancer in small image patches taken from larger digital pathology scans on histopathological images [19]. InstantDL achieved an AUC of 0.93 while the winner of the challenge achieved an AUC of 1.0 and the median participant 0.91

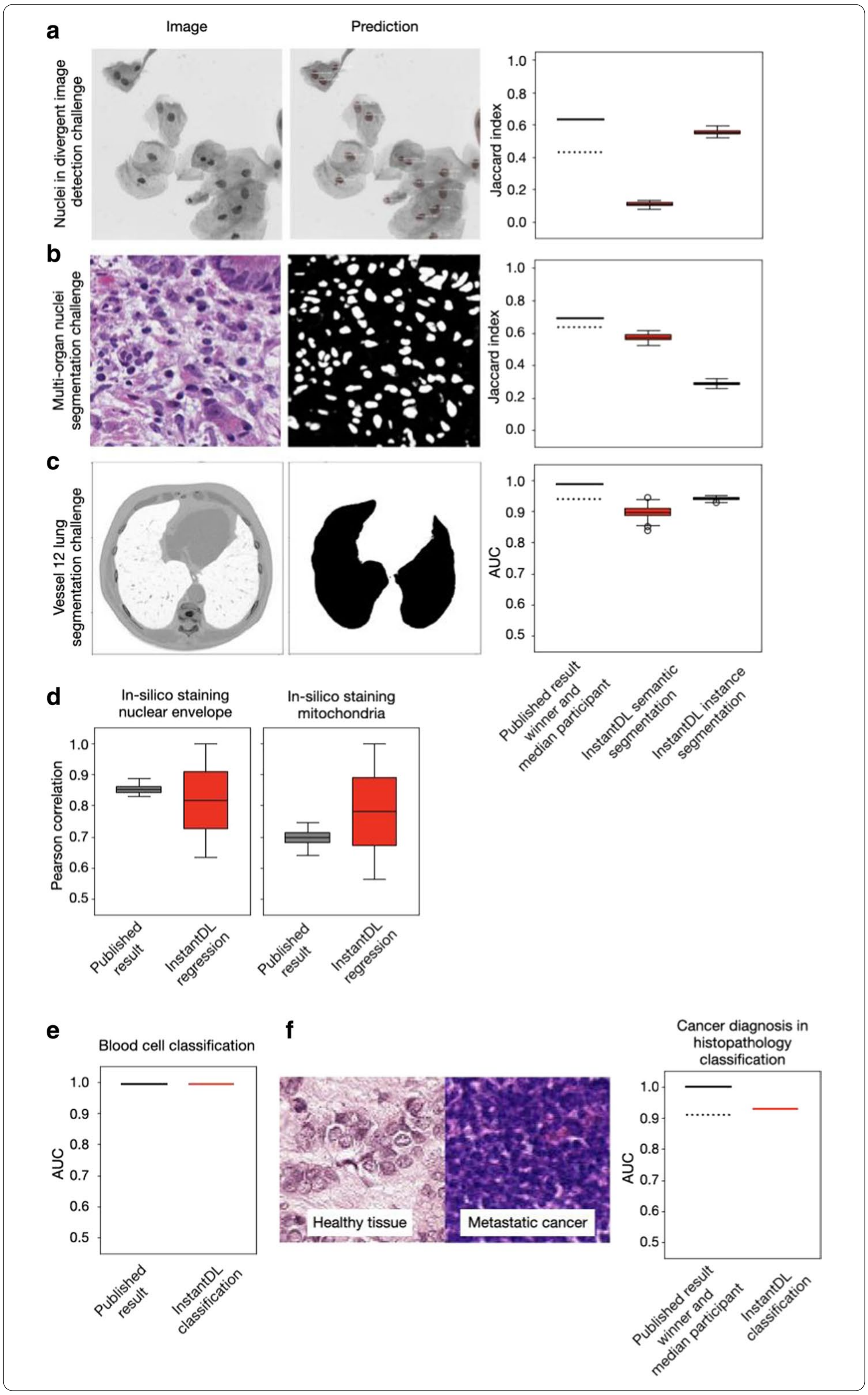
segmentation with Google-Colab GPU took 9 min. This dataset contains 120 training images of size 512 × 512 pixels and we ran it with a batch size of 1 for 37 epochs.

Results

To evaluate InstantDL broadly, we applied it to seven publically available datasets (four of which come from data science challenges) and compared its performance to published results. If no test set was provided, we took 20% of the data to create our own test set. This was done on the highest level of abstraction, for example on patient level or tissue-slide level whenever possible, otherwise the data was randomly split. We used the same evaluation metrics as published in the respective papers (Jaccard index, AUC, Pearson correlation) to compare our results appropriately.

For pre-processing, we transformed the images to*.tiff* files and classification labels to a*.csv* file to adapt them to the InstantDL requirements. Training was performed by saving the best model using early stopping. As data augmentation we used horizontal and vertical flipping. For pixel-wise regression we used mean squared error loss, for semantic segmentation we used binary cross entropy loss and for classification we used categorical cross entropy loss. For instance segmentation, binary cross-entropy was used as segmentation loss in combination with the localization and classification loss in the Mask-RCNN [29].

We evaluated the performance of semantic segmentation and instance segmentation on three datasets. In the first dataset we segmented nuclei in microscopy images contained in the Data Science Bowl 2018 [13] dataset. Using InstandDL instance segmentation we reached a median Jaccard index of 0.60 (25–75%ile: 0.61 to 0.58 estimated from bootstrapping), while using semantic segmentation we reached a median Jaccard index of 0.16 (25–75%ile: 0.15 to 0.17). The winner of the challenge reached a Jaccard


a
Image
Prediction
Nuclei in divergent image detection challenge
Jaccard index
1.0
0.8
0.6
0.4
0.2
0.0
b
Multi-organ nuclei segmentation challenge
Jaccard index
c
Vessel 12 lung segmentation challenge
AUC
1.0
0.9
0.8
0.7
0.6
0.5
Published result winner and median participant
InstantDL semantic segmentation
InstantDL instance segmentation
d
In-silico staining nuclear envelope
In-silico staining mitochondria
Pearson correlation
Published result
InstantDL regression
e
Blood cell classification
AUC
Published result
InstantDL classification
f
Healthy tissue
Metastatic cancer
Cancer diagnosis in histopathology classification
AUC
Published result winner and median participant
InstantDL classification

index of 0.63 while the median participant reached 0.42 (solid and dotted line, Fig. 3a). The second task was the multi-organ nuclei segmentation challenge. Here, 30 microscopy images of various organs with hematoxylin and eosin staining are provided [14, 15]. We reached a median Jaccard score of 0.57 (25–75%ile: 0.56 to 0.59) with InstantDL's semantic segmentation and 0.29 (25–75%ile: 0.28 to 0.30) with instance segmentation. The winner of the challenge reached 0.69 and the median participant scored 0.63 (solid and dotted line, Fig. 3b). Thirdly, we benchmarked InstantDL on lung CT images from the Vessel-12 challenge [16]. Using instance segmentation we reached an area under the receiver operator curve (AUC) of 0.94 (25–75%ile: 0.94 to 0.94), and 0.90 (25–75%ile: 0.88 to 0.92) with semantic segmentation. The winner of the challenge reached a score of 0.99 and the median participant 0.94 (solid and dotted line, Fig. 3c).

To evaluate InstantDL's performance for pixel-wise regression, we predicted the 3D nuclear envelope and mitochondria staining from brightfield images [7]. For the nuclear envelope staining prediction, we achieved a median pixel-wise Pearson correlation to the real staining of 0.85 and 0.78 for the prediction of mitochondria staining (Fig. 3d), similar to the published result.

Classification performance was evaluated on two datasets. In Matek et al. images of single white blood cells from 100 leukemic and 100 non-leukemic individuals were classified into leukemic blast cell images vs. benign white blood cell subtype images [6]. We reached an AUC of 0.99 where Matek et al. reached 0.99 (Fig. 3e). The second task was to classify metastatic cancer image patches of digital pathology scans, taken from the Histopathologic Cancer Detection Kaggle challenge [19]. We reached an AUC of 0.93, while the winner reached 1.00 (solid line in Fig. 3f) and the median participant scored 0.91 (dotted line, Fig. 3f).

Discussion

We present InstantDL, a deep learning pipeline for semantic segmentation, instance segmentation, pixel-wise regression and classification of biomedical images. InstantDL simplifies the access to the advantages of deep learning algorithms for biomedical researchers with limited computer science background. The only requirement is a solid understanding of the data (and how appropriately split it into training and test set), as well as of the task and loss function that should be optimized during training the model (see e.g. [1, 43]). The pipeline is designed for maximum automation to make training and testing as convenient and as easy as possible. However, some parameter settings depend on the dataset properties and therefore cannot be automated. After setting a maximum of 11 parameters, the pipeline can be run without further user interactions. We included state-of-the-art analysis metrics that are accessible out of the box. Moreover, we included uncertainty prediction to provide an additional level of interpretability of predictions.

We tested the performance of InstantDL on a variety of publicly available datasets and achieved competitive results without any hyperparameter tuning. While we could not reach the winners' performance (typically using elaborate problem specific algorithms and data pre- and post-processing) of the respective image computing challenges with our out-of-the-box approach, the output of InstantDL suffices for standard biomedical data analytics performed after the image processing step.

Table 2 Comparison of InstantDL to other deep learning frameworks

	InstantDL	Open ML	ImJoy	ZeroCostDL4Mic
Host	Local, on cluster, or Google-Colab	Web based	Web based platform	Web based (Google-Colab)
Data privacy	Yes (running locally)	No (shared with upload)	Limited (hosted in the cloud)	Limited (hosted in the cloud)
Target audience	Researchers and developers	Researchers and developers	Biomedical researchers	Biomedical researchers
Developed for	Biomedical images	All kinds of data	All kinds of data	Biomedical images
Customizability of Code	Open source	Open source	Open source	Open source
Cost	Free	Free	Free	Free

To improve performance on a specific dataset we recommend to select specialized data augmentations and a suitable loss function in the configuration file for higher performance. Expert users can also adapt InstantDL's code to their needs. We plan to extend InstantDL by implementing self-supervised and semi-supervised learning methods to utilize unlalleded data optimally in the future [34]. To improve semantic segmentation specifically, we will implement specialized loss functions, such as Malis loss [46]

The networks currently implemented are suitable for the most frequently used image processing tasks. Due to InstantDLs modular implementation it is however easy for users with python knowledge to exchange deep learning algorithms for a taylormade solution, e.g. other classifications networks like the ResNext [44] or MobileNet [45].

Other deep learning frameworks such as OpenML [46], ImJoy [47] and ZeroCostDL4Mic [48], nucleAlzer [3] and yeastspotter [20] allow for the application of deep learning methods on user data (Table 2). However, they all require data upload to a cloud system. OpenML, e.g., offers an online ecosystem of datasets and machine learning models, but the dataset will be made publically available with upload. The web tools nucleAlzer [3] and yeastspotter [20] are extremely easy to use, but provide only cell and nuclei segmentation using drag-and-drop with pre-trained machine learning models. ImJoy [47] and ZeroCostDL4Mic [48] are more flexible frameworks. ImJoy is an online tool where users can select plugins for processing their data, which is easy and convenient, yet not locally executable. Users can create their own plugins requiring professional programming knowledge. ZeroCostDL4Mic requires similar knowledge about programming as InstantDL, but solely relies on Google Colab. This requires the upload of data to the users Google drive, which has a memory limit of 15 GB in the free version and a runtime limit of 12 h. This can impose a hurdle for users, in particular for large datasets, common e.g. in computational pathology applications. InstantDL is applicable to any 2D or 3D image segmentation task and the only tool offering uncertainty estimation. Our pipeline can easily be installed and run locally on a computer or server, ensuring data scalability, privacy and security. However, InstantDL can also be used on cloud solutions, such as Google Colab. Table 2 compares the features of InstantDL with other frameworks.

Conclusions

InstantDLs code is publicly available and convenient to tune and extend. It is ideal for labs looking for a customizable tool to apply deep learning to their own data. We thus hope to empower biomedical researchers to conduct reproducible image processing with a convenient and easy-to-use pipeline.

Availability and requirements

Project name: InstantDL.
Project home page: https://github.com/marrlab/InstantDL
Operating system(s): Platform independent.
Programming language: Python.
Other requirements: cudatoolkit: 10.1.243 # in case of GPU existence, cudnn: 7.6.5 # in case of GPU existence, h5py: 2.9.0, hdf5: 1.10.4, imageio: 2.6.1, keras: 2.2.4, matplotlib: 3.1.1, numpy: 1.16.4, python: 3.6.7, scikit-image: 0.15.0, scikit-learn: 0.21.3, scipy: 1.3.0, tensorboard: 1.14.0, tensorflow: 1.14.0, tensorflow-gpu: 1.14.0 # in case of GPU existence, pandas: 1.0.3
License: MIT.
Any restrictions to use by non-academics: None.
For reproducing our results, all data used for illustration and benchmarking is available for download at https://hmgubox2.helmholtz-muenchen.de/index.php/s/YXRD4a7qHnCa9x5.

Availability and implementation

InstantDL is available under the terms of MIT licence. It can be found on GitHub: https://github.com/marrlab/InstantDL

Supplementary informationWeights and validation data are available under: https://hmgubox2.helmholtz-muenchen.de/index.php/s/YXRD4a7qHnCa9x5

Abbreviations

ROC: Receiver operating characteristic; AUC: Area under curve.

Acknowledgements

We thank Niklas Köhler and Nikos Chlis (Munich) for contributing to the initial concept for InstantDL. We thank Daniel Schirmacher (Zürich), Lea Schuh, Johanna Winter, Moritz Thomas, Matthias Hehr, Benjamin Schubert, Niklas Kiermeyer, and Benedikt Mairhörmann (all Munich) for valuable feedback on the manuscript and using InstantDL.

Authors' contributions

DW implemented the pipeline and conducted experiments. He wrote the manuscript and created the figures with CM and SSB. SSB revised and refactored the code, added docker installation and tested the pipeline. CM supervised the study. All authors read and approved the manuscript.

Funding

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 862811 (RSENSE). CM was supported by the BMBF, grant 01ZX1710A-F (Micmode-I2T) and has received funding from the European Research Council (ERC) under the European Union's Horizon 2020 research and innovation programme (Grant agreement No. 866411). SSB is a member of the Munich School for Data Science (MUDS). The funding bodys did not play any roles in the design of the study and collection, analysis, and interpretation of data and in writing the manuscript.

Availability of data and materials

The datasets generated and/or analysed during the current study are available in the HGMU repository, https://hmgubox2.helmholtz-muenchen.de/index.php/s/YXRD4a7qHnCa9x5 and the GitHub repository, https://github.com/marrlab/InstantDL

Ethics approval and consent to participate

Not applicable.

Consent for publication

Not applicable.

Competing interests

The authors declare that they have no competing interests.

Author details

[1] Institute of Computational Biology, Helmholtz Zentrum München - German Research Center for Environmental Health, Neuherberg, Germany. [2] School of Life Sciences, Technical University of Munich, Weihenstephan, Germany. [3] Roche Innovation Center Munich, Roche Diagnostics GmbH, Penzberg, Germany.

Received: 6 October 2020 Accepted: 21 February 2021
Published online: 02 March 2021

References

1. LeCun Y, Bengio Y, Hinton G. Deep learning. Nature. 2015;521:436–44.
2. Caicedo JC, Roth J, Goodman A, Becker T, Karhohs KW, Broisin M, Molnar C, McQuin C, Singh S, Theis FJ, Carpenter AE. Evaluation of deep learning strategies for nucleus segmentation in fluorescence images. Cytometry A. 2019;95:952–65.
3. Hollandi R, Szkalisity A, Toth T, Tasnadi E, Molnar C, Mathe B, Grexa I, Molnar J, Balind A, Gorbe M, Kovacs M, Migh E, Goodman A, Balassa T, Koos K, Wang W, Caicedo JC, Bara N, Kovacs F, Paavolainen L, Danka T, Kriston A, Carpenter AE, Smith K, Horvath P. nucleAIzer: a parameter-free deep learning framework for nucleus segmentation using image style transfer. Cell Systems. 2020;10:453–8.
4. Cireşan DC, Giusti A, Gambardella LM, Schmidhuber J. Mitosis detection in breast cancer histology images with deep neural networks. Med Image Comput Comput Assist Interv. 2013;16:411–8.
5. Buggenthin F, Buettner F, Hoppe PS, Endele M, Kroiss M, Strasser M, Schwarzfischer M, Loeffler D, Kokkaliaris KD, Hilsenbeck O, Schroeder T, Theis FJ, Marr C. Prospective identification of hematopoietic lineage choice by deep learning. Nat Methods. 2017;14:403–6.
6. Matek C, Schwarz S, Spiekermann K, Marr C. Human-level recognition of blast cells in acute myeloid leukaemia with convolutional neural networks. Nat Mach Intell. 2019;1:538–44.
7. Ounkomol C, Seshamani S, Maleckar MM, Collman F, Johnson G. Label-free prediction of three-dimensional fluorescence images from transmitted light microscopy. bioRxiv 2018;289504.
8. Christiansen EM, Yang SJ, Michael Ando D, Javaherian A, Skibinski G, Lipnick S, Mount E, O'Neil A, Shah K, Lee AK, Goyal P, Fedus W, Poplin R, Esteva A, Berndl M, Rubin LL, Nelson P, Finkbeiner S. In Silico labeling: predicting fluorescent labels in unlabeled images. Cell. 2018;173:792–803.
9. Esteva A, Kuprel B, Novoa RA, Ko J, Swetter SM, Blau HM, Thrun S. Dermatologist-level classification of skin cancer with deep neural networks. Nature. 2017;542:115–8.
10. McKinney SM, Sieniek M, Godbole V, Godwin J, Antropova N, Ashrafian H, Back T, Chesus M, Corrado GC, Darzi A, Etemadi M, Garcia-Vicente F, Gilbert FJ, Halling-Brown M, Hassabis D, Jansen S, Karthikesalingam A, Kelly CJ, King D, Ledsam JR, Melnick D, Mostofi H, Peng L, Reicher JJ, Romera-Paredes B, Sidebottom R, Suleyman M, Tse D, Young KC, De Fauw J, Shetty S. International evaluation of an AI system for breast cancer screening. Nature. 2020;577:89–94.
11. Tschandl P, Codella N, Akay BN, Argenziano G, Braun RP, Cabo H, Gutman D, Halpern A, Helba B, Hofmann-Wellenhof R, Lallas A, Lapins J, Longo C, Malvehy J, Marchetti MA, Marghoob A, Menzies S, Oakley A, Paoli J, Puig S, Rinner C, Rosendahl C, Scope A, Sinz C, Soyer HP, Thomas L, Zalaudek I, Kittler H. Comparison of the accuracy of human readers versus machine-learning algorithms for pigmented skin lesion classification: an open, web-based, international, diagnostic study. Lancet Oncol. 2019;20:938–47.
12. Liu X, Faes L, Kale AU, Wagner SK, Fu DJ, Bruynseels A, Mahendiran T, Moraes G, Shamdas M, Kern C, Ledsam JR, Schmid MK, Balaskas K, Topol EJ, Bachmann LM, Keane PA, Denniston AK. A comparison of deep learning performance against health-care professionals in detecting diseases from medical imaging: a systematic review and meta-analysis. Lancet Digit Health. 2019;1:e271–97.
13. Caicedo JC, Goodman A, Karhohs KW, Cimini BA, Ackerman J, Haghighi M, Heng C, Becker T, Doan M, McQuin C, Rohban M, Singh S, Carpenter AE. Nucleus segmentation across imaging experiments: the 2018 Data Science Bowl. Nat Methods. 2019;16:1247–53.
14. Kumar N, Verma R, Anand D, Zhou Y, Onder OF, Tsougenis E, Chen H, Heng P-A, Li J, Hu Z, Wang Y, Koohbanani NA, Jahanifar M, Tajeddin NZ, Gooya A, Rajpoot N, Ren X, Zhou S, Wang Q, Shen D, Yang C-K, Weng C-H, Yu W-H, Yeh C-Y, Yang S, Xu S, Yeung PH, Sun P, Mahbod A, Schaefer G, Ellinger I, Ecker R, Smedby O, Wang C, Chidester B, Ton T-V, Tran M-T, Ma J, Do MN, Graham S, Vu QD, Kwak JT, Gunda A, Chunduri R, Hu C, Zhou X, Lotfi D, Safdari R, Kascenas A, O'Neil A, Eschweiler D, Stegmaier J, Cui Y, Yin B, Chen K, Tian X, Gruening P, Barth E, Arbel E, Remer I, Ben-Dor A, Sirazitdinova E, Kohl M, Braunewell S, Li Y, Xie X, Shen L, Ma J, Baksi KD, Khan MA, Choo J, Colomer A, Naranjo V, Pei L, Iftekharuddin KM, Roy K, Bhattacharjee D, Pedraza A, Bueno MG, Devanathan S, Radhakrishnan S, Koduganty P, Wu Z, Cai G, Liu X, Wang Y, Sethi A. A multi-organ nucleus segmentation challenge. IEEE Trans Med Imaging. 2020;39:1380–91.

15. Kumar N, Verma R, Sharma S, Bhargava S, Vahadane A, Sethi A. A dataset and a technique for generalized nuclear segmentation for computational pathology. IEEE Trans Med Imaging. 2017;36:1550–60.
16. Rudyanto RD, Kerkstra S, van Rikxoort EM, Fetita C, Brillet P-Y, Lefevre C, Xue W, Zhu X, Liang J, Öksüz I, Ünay D, Kadipaşaoğlu K, Estépar RSJ, Ross JC, Washko GR, Prieto J-C, Hoyos MH, Orkisz M, Meine H, Hüllebrand M, Stöcker C, Mir FL, Naranjo V, Villanueva E, Staring M, Xiao C, Stoel BC, Fabijanska A, Smistad E, Elster AC, Lindseth F, Foruzan AH, Kiros R, Popuri K, Cobzas D, Jimenez-Carretero D, Santos A, Ledesma-Carbayo MJ, Helmberger M, Urschler M, Pienn M, Bosboom DGH, Campo A, Prokop M, de Jong PA, Ortiz-de-Solorzano C, Muñoz-Barrutia A, van Ginneken B. Comparing algorithms for automated vessel segmentation in computed tomography scans of the lung: the VESSEL12 study. Med Image Anal. 2014;18:1217–32.
17. Matek C, Schwarz S, Spiekermann K, Marr C. Human-level recognition of blast cells in acute myeloid leukemia with convolutional neural networks. bioRxiv 2019.
18. Shen L, Margolies LR, Rothstein JH, Fluder E, McBride R, Sieh W. Deep learning to improve breast cancer detection on screening mammography. Sci Rep. 2019;9:12495.
19. Histopathologic Cancer Detection. https://www.kaggle.com/c/histopathologic-cancer-detection/overview.
20. Lu AX, Zarin T, Hsu IS, Moses AM. YeastSpotter: accurate and parameter-free web segmentation for microscopy images of yeast cells. Bioinformatics. 2019;35:4525–7.
21. Litjens G, Kooi T, Bejnordi BE, Setio AAA, Ciompi F, Ghafoorian M, van der Laak JAWM, van Ginneken B, Sánchez CI. A survey on deep learning in medical image analysis. Med Image Anal. 2017;42:60–88.
22. Maier A, Syben C, Lasser T, Riess C. A gentle introduction to deep learning in medical image processing. Z Med Phys. 2019;29:86–101.
23. Minaee S, Boykov Y, Porikli F, Plaza A, Kehtarnavaz N, Terzopoulos D. Image segmentation using deep learning: a survey. arXiv:2001.05566 [cs.CV] (2020).
24. Ronneberger O, Fischer P, Brox T. UNet convolutional networks for biomedical image segmentation. arXiv:1505.04597 [cs.CV] (2015).
25. Falk T, Mai D, Bensch R, Çiçek Ö, Abdulkadir A, Marrakchi Y, Böhm A, Deubner J, Jäckel Z, Seiwald K, Dovzhenko A, Tietz O, Dal Bosco C, Walsh S, Saltukoglu D, Tay TL, Prinz M, Palme K, Simons M, Diester I, Brox T, Ronneberger O. U-Net: deep learning for cell counting, detection, and morphometry. Nat Methods. 2019;16:67–70.
26. Isensee F, Petersen J, Klein A, Zimmerer D, Jaeger PF, Kohl S, Wasserthal J, Koehler G, Norajitra T, Wirkert S, Maier-Hein KH. nnU-Net: self-adapting framework for U-Net-based medical image segmentation. arXiv:1809.10486 [cs.CV] (2018).
27. Otsu N. A threshold selection method from gray-level histograms. IEEE Trans Syst Man Cybern. 1979;9:62–6.
28. Caicedo JC, Cooper S, Heigwer F, Warchal S, Qiu P, Molnar C, Vasilevich AS, Barry JD, Bansal HS, Kraus O, Wawer M, Paavolainen L, Herrmann MD, Rohban M, Hung J, Hennig H, Concannon J, Smith I, Clemons PA, Singh S, Rees P, Horvath P, Linington RG, Carpenter AE. Data-analysis strategies for image-based cell profiling. Nat Methods. 2017;14:849–63.
29. Kaiming H, Georgia G, Piotr D, Ross G. Mask r-cnn. Vis: Proc. IEEE Int. Conf. Comput; 2017.
30. Abdulla W. Mask R-CNN for object detection and instance segmentation on Keras and TensorFlow. GitHub repository 2017.
31. He K, Zhang X, Ren S, Sun J. Deep residual learning for image recognition. In Proceedings of the IEEE conference on computer vision and pattern recognition. 2016. p. 770–8.
32. Brinker TJ, Hekler A, Utikal JS, Grabe N, Schadendorf D, Klode J, Berking C, Steeb T, Enk AH, von Kalle C. Skin cancer classification using convolutional neural networks: systematic review. J Med Internet Res. 2018;20:e11936.
33. Waibel D, Gröhl J, Isensee F, Kirchner T, Maier-Hein K, Maier-Hein L. Reconstruction of initial pressure from limited view photoacoustic images using deep learning. In: Photons plus ultrasound: imaging and sensing 2018. International Society for Optics and Photonics. 2018;10494:104942S.
34. Boushehri SS, Qasim AB, Waibel D, Schmich F, Marr C. Annotation-efficient classification combining active learning, pre-training and semi-supervised learning for biomedical images. Cold Spring Harbor Laboratory 2020:2020.12.07.414235.
35. Shorten C, Khoshgoftaar TM. A survey on image data augmentation for deep learning. J Big Data. 2019;6:60.
36. Kingma DP, Ba J. Adam: a method for stochastic optimization. arXiv:1412.6980 [cs.LG] (2014)
37. Gal Y, Ghahramani Z. Dropout as a Bayesian approximation: representing model uncertainty in deep learning. In: International conference on machine learning. 2016. p. 1050–9.
38. Lakshminarayanan B, Pritzel A, Blundell C. Simple and Scalable predictive uncertainty estimation using deep ensembles. In: Guyon I, Luxburg UV, Bengio S, Wallach H, Fergus R, Vishwanathan S, Garnett R, editors. Advances in neural information processing systems 30. Curran Associates, Inc.; 2017. p. 6402–13.
39. Keras. https://keras.io.
40. Bisong E. Google Colaboratory. In: Bisong E, editor. Building machine learning and deep learning models on google cloud platform: a comprehensive guide for beginners. Berkeley, CA: Apress; 2019. p. 59–64.
41. Deep Learning to Detect Skin Cancer using Google Colab. Ijeat 2019, 8:2176–2183.
42. Docker Documentation. https://docs.docker.com/.
43. Goodfellow I, Bengio Y, Courville A. Deep learning. Cambridge: MIT Press; 2016.
44. Xie S, Girshick R, Dollár P, Tu Z, He K. Aggregated residual transformations for deep neural networks. In Proceedings of the IEEE conference on computer vision and pattern recognition. 2017. p. 1492–1500.
45. Howard AG, Zhu M, Chen B, Kalenichenko D, Wang W, Weyand T, Andreetto M, Adam H. MobileNets: efficient convolutional neural networks for mobile vision applications. arXiv:1704.04861 [cs.CV] (2017).
46. Vanschoren J, van Rijn JN, Bischl B, Torgo L. OpenML: networked science in machine learning. SIGKDD Explor. 2013;15(2):49–60.
47. Ouyang W, Mueller F, Hjelmare M, Lundberg E, Zimmer C. ImJoy: an open-source computational platform for the deep learning era. Nat Methods. 2019;16:1199–200.
48. Von Chamier L, Jukkala J, Spahn C, Lerche M. ZeroCostDL4Mic: an open platform to simplify access and use of deep-learning in microscopy. BioRxiv 2020.

Publisher's Note

Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

A.2. SHAPR - An AI approach to predict 3D cell shapes from 2D microscopic images

Dominik J. E. Waibel, Niklas Kiermeyer, Scott Atwell, Ario Sadafi, Matthias Meier, and Carsten Marr

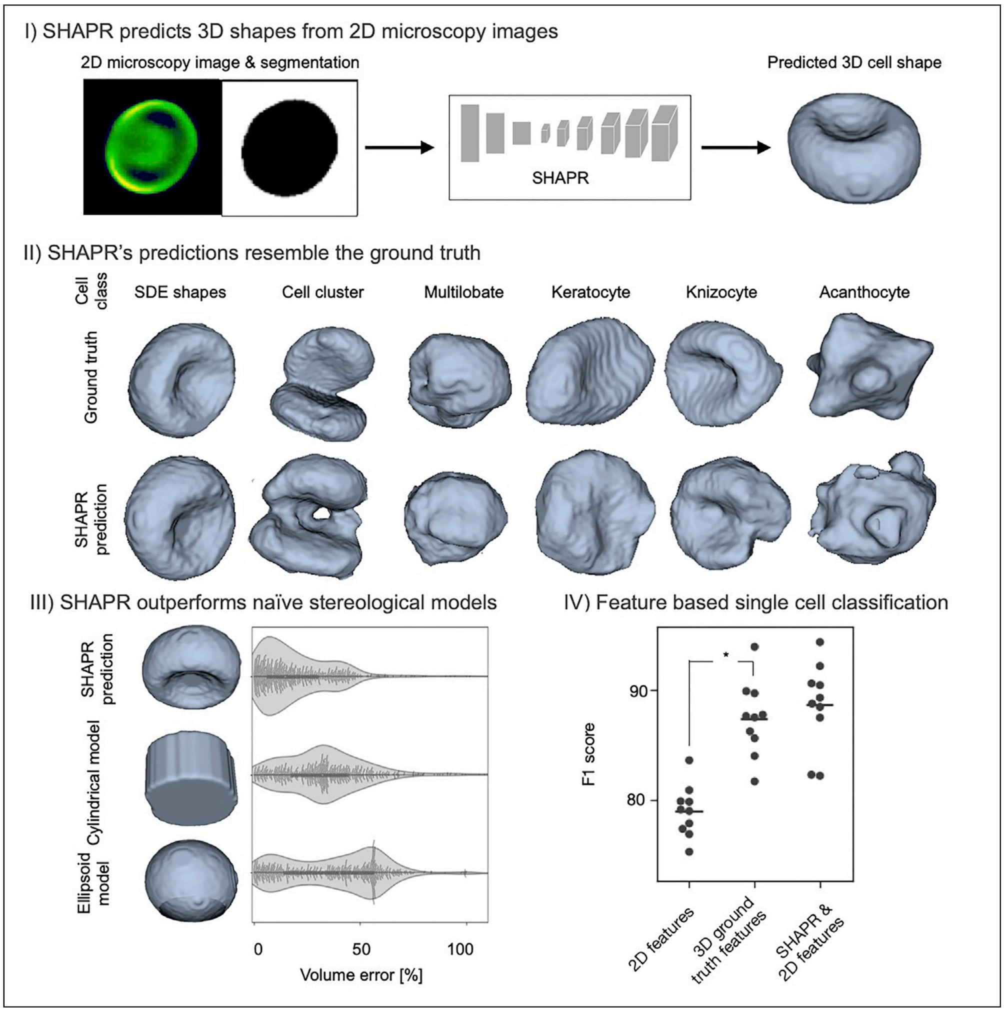
This is a published version of the article in iScience following peer review. The article is open-access thus the published version is inserted here.
This article is published in iScience under the Creative Commons Attribution license (CC BY-NC-ND 4.0).

iScience

CellPress OPEN ACCESS

Article

SHAPR predicts 3D cell shapes from 2D microscopic images



Dominik J.E. Waibel, Niklas Kiermeyer, Scott Atwell, Ario Sadafi, Matthias Meier, Carsten Marr

matthias.meier@helmholtz-muenchen.de (M.M.)
carsten.marr@helmholtz-muenchen.de (C.M.)

Highlights

SHAPR predicts 3D single cell shapes from 2D microscopic images

It is trained with a two-step supervised and adversarial approach

SHAPR improves morphological feature based cell classification

SHAPR learns fundamental 3D shape properties of human-induced pluripotent stem cells

Waibel et al., iScience 25, 105298
November 18, 2022 © 2022 The Authors.
https://doi.org/10.1016/j.isci.2022.105298

iScience

CellPress
OPEN ACCESS

Article

SHAPR predicts 3D cell shapes from 2D microscopic images

Dominik J.E. Waibel,[1,2,3] Niklas Kiermeyer,[1,2] Scott Atwell,[5] Ario Sadafi,[1,2,4] Matthias Meier,[5,*] and Carsten Marr[1,2,6,*]

SUMMARY

Reconstruction of shapes and sizes of three-dimensional (3D) objects from two-dimensional (2D) information is an intensely studied subject in computer vision. We here consider the level of single cells and nuclei and present a neural network-based SHApe PRediction autoencoder. For proof-of-concept, SHAPR reconstructs 3D shapes of red blood cells from single view 2D confocal microscopy images more accurately than naïve stereological models and significantly increases the feature-based prediction of red blood cell types from F1 = 79% to F1 = 87.4%. Applied to 2D images containing spheroidal aggregates of densely grown human induced pluripotent stem cells, we find that SHAPR learns fundamental shape properties of cell nuclei and allows for prediction-based morphometry. Reducing imaging time and data storage, SHAPR will help to optimize and up-scale image-based high-throughput applications for biomedicine.

INTRODUCTION

Recording single cells with confocal microscopy for high-throughput biomedical applications is prohibitively time-consuming in three dimensions (3D) as it requires the acquisition of multiple two-dimensional (2D) images. This raises the question of how to optimally trade-off between throughput and resolution in space and time. A number of methods to reduce imaging time for single cell characterization have recently been developed, ranging from microscopic techniques, such as optical diffraction tomography (Sung et al., 2009), digital holographic imaging (Javidi et al.) or integral imaging microscopy (Moon et al., 2013; Martínez-Corral and Javidi, 2018), to in-silico fluorescence staining (Ounkomol et al., 2018; Christiansen et al., 2018; Rivenson et al., 2019) and image restoration techniques (Weigert et al., 2018). To predict single-cell 3D morphology, one would ideally be able to exploit the information in 2D fluorescence microscopy images. Deep learning-based solutions for predicting 3D object shapes from photographs exist, creating meshes (Wang et al., 2018; Gkioxari et al., 2019), voxels (Choy et al., 2016), or point clouds (Fan et al., 2017) for airplanes, cars, and furniture, but they cannot be translated to fluorescence microscopy for several reasons. First, fluorescence microscopy imaging is fundamentally different from real-world photographs in terms of color, contrast, and object orientation. Second, unlike the shapes of cars or furniture that might vary because of differing photographic viewpoints, the shapes of single cells are similar but never the same, and it is often not feasible to image the same cell from different angles in high throughput microscopy. Finally, existing computer vision algorithms have been trained on tens of thousands of photographs where synthetic 3D models are available (Chang et al., 2015; Sun et al., 2018; Xiang et al., 2014). In the biomedical domain the number of potential training images is orders of magnitude smaller. Although Wu et al. (2019) have demonstrated that neural networks can be used for 3D refocusing onto a user-defined surface from 2D microscopy images containing single fluorescent beads or fluorophore signals, to the best of our knowledge no model exists in the biomedical domain to reconstruct 3D cell shapes from 2D confocal microscopy images.

RESULTS

We addressed these problems with SHAPR, a deep learning network algorithm that combines a 2D encoder for feature extraction from 2D images with a 3D decoder to predict 3D shapes from a latent space representation (Figures 1A and S1).

[1]Institute of AI for Health, Helmholtz Munich - German Research Center for Environmental Health, Neuherberg, Germany

[2]Institute of Computational Biology, Helmholtz Munich - German Research Center for Environmental Health, Neuherberg, Germany

[3]Technical University of Munich, School of Life Sciences, Weihenstephan, Germany

[4]Computer Aided Medical Procedures, Technical University of Munich, Munich, Germany

[5]Helmholtz Pioneer Campus, Helmholtz Munich - German Research Center for Environmental Health, Neuherberg, Germany

[6]Lead contact

*Correspondence: matthias.meier@helmholtz-muenchen.de (M.M.), carsten.marr@helmholtz-muenchen.de (C.M.)

https://doi.org/10.1016/j.isci.2022.105298

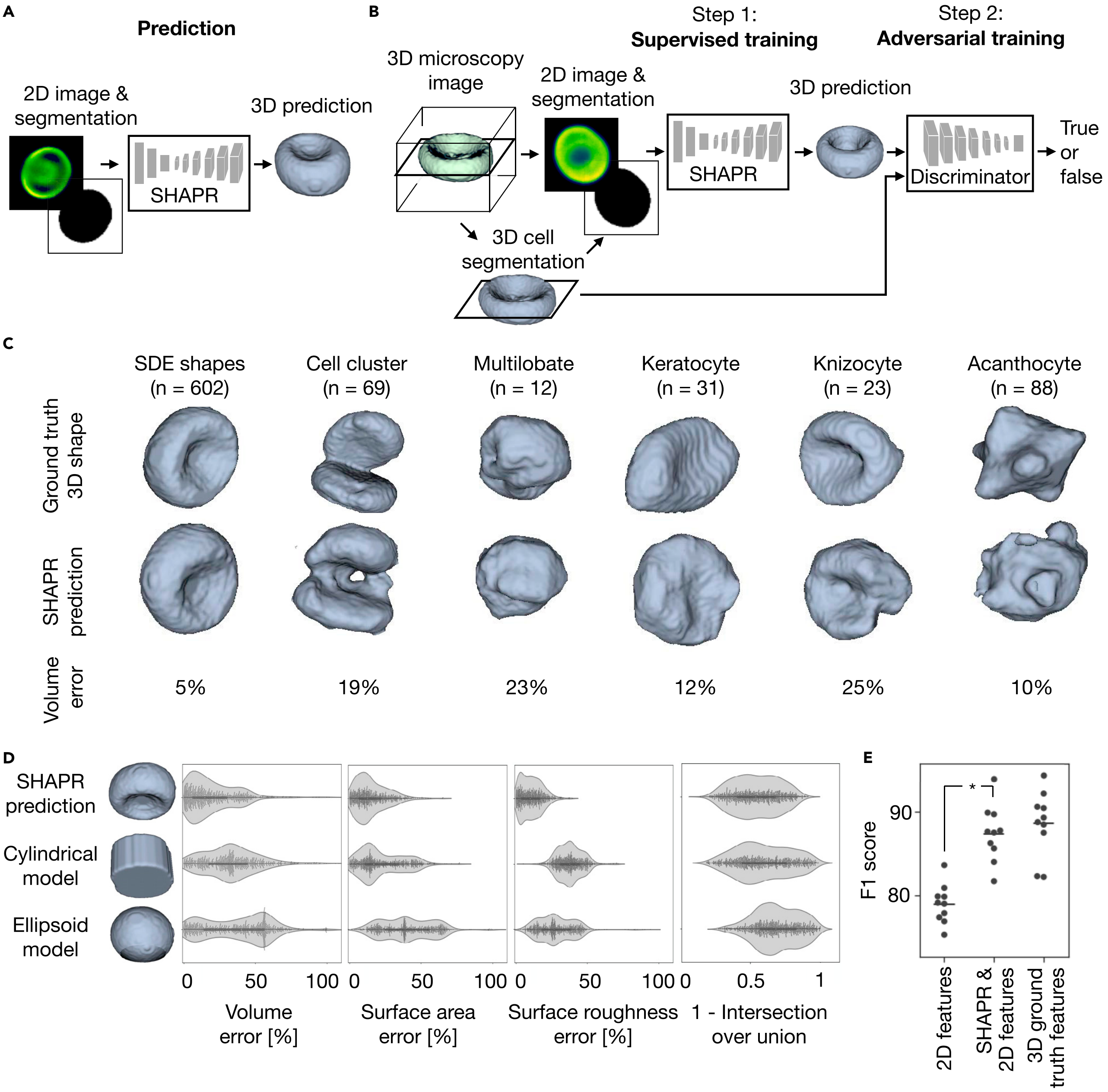


Figure 1. SHAPR predicts 3D cell shapes from 2D microscopic images more accurately than naïve stereological models and improves shape classification

(A) SHAPR consists of an encoder for embedding 2D images into a 128-dimensional latent space and a decoder for reconstructing 3D cell shapes from the latent space representations.

(B) Two-step training approach: during step 1, SHAPR was trained in a supervised fashion with 2D fluorescent confocal cell microscopy images and their corresponding binary segmentations from a red blood cell library. During step 2, SHAPR was fine-tuned with a discriminator challenging its cell shape predictions.

(C) Example predictions for a set of red blood cells representing six different classes. The SDE shape class combines spherocytes, stomatocytes, discocytes, and echinocytes.

(D) The volume error is significantly lower for SHAPR (20 ± 18%) as compared to two naïve stereological models ($\text{Volume}_{\text{cylindrical}} = 33 \pm 22$, $p_{\text{cylindrical}} = 2.6 \times 10^{-46}$; and $\text{Volume}_{\text{ellipsoid}} = 37 \pm 23$; $p_{\text{ellipsoid}} = 7.8 \times 10^{-73}$, n = 825, paired Wilcoxon signed-rank test). Volume, surface area, and roughness error are significantly reduced.

(E) Random forest-based red blood cell classification is significantly improved when morphological features extracted from SHAPR predicted cell shapes are added to features derived from 2D images (p = 0.005, paired Wilcoxon signed-rank test, n = 825).

SHAPR predicts 3D cell shapes from 2D microscopic images more accurately than naïve stereological models

For proof of concept, we predicted cell shapes using a recently published library detailing 3D red blood cell shapes (n = 825 cells) (Simionato et al., 2021). Each cell shape was reconstructed from 68 confocal images with a z-resolution of 0.3μm. Using the 2D image that intersects the red blood cell at the center slice, and the corresponding segmentation as input, SHAPR was trained by minimizing binary cross-entropy and dice loss between the true and the predicted 3D red blood cell shape (Figure 1B and STAR methods). To increase SHAPR's prediction accuracy, a discriminator model (Figure 1B) was trained to differentiate between true and predicted 3D cell shapes. SHAPR and the discriminator were trained until the predicted cell shape converged to an optimum. In each one of five cross-validation runs, 495 (60%) red blood cells from the library were used for training and 165 (20%) for intermediate validation during training. During testing, we predicted the 3D shapes of 165 (20%) previously unseen red blood cells. The results demonstrate that SHAPR is able to predict single red blood cell 3D shapes: although non-complex morphologies from red blood cells with a biconcave discoid shape (stomatocyte-discocyte-echinocyte (SDE) shape class) were predicted with low relative volume error, more complex shapes with irregular protrusions or cavities as seen in knizocytes and acanthocytes were predicted with larger errors (Figure 1C). We compared this cell shape prediction to two naïve stereological models, i.e., a cylindrical and an ellipsoid fit (see STAR methods). SHAPR predictions significantly outperformed these models with respect to volume error, 2D surface area error, surface roughness error and intersection over union (Figure 1D).

SHAPR improves morphological feature based cell classification

Simionato et al. (Simionato et al., (2021) classified red blood cells into six categories (Figure 1C) based on their 3D morphology. Can SHAPR predictions from 2D images improve such a downstream classification task? To investigate this, we extracted 126 morphological features, 11 features from an additionally predicted object mesh, object moments up to third order, correlation and dissimilarity of gray level co-occurrence matrices, and 64 Gabor features (see STAR methods for more details) from each predicted 3D cell shape. Using random forest (Breiman, 2001), we classified each blood cell into one of the six classes and compared SHAPR's performance with the 3D ground truth features and a 2D baseline, where only features derived from the 2D image and segmentation were used (see STAR methods). As expected, classification based on ground truth features led to the highest F1 score (Figure 1E; 88.6 ± 3.7%). Strikingly, enriching 2D features with SHAPR derived features performed significantly better (F1 = 87.4 ± 3.1%, mean ± std.dev., n = 10 cross-validation runs) than using 2D features only (F1 = 79.0 ± 2.2%) in a tenfold cross-validation (Figure 1E, p = 0.005, paired Wilcoxon signed-rank test).

SHAPR learns fundamental 3D shape properties of human-induced pluripotent stem cell (iPSC) nuclei from a single 2D slice

Predicting shapes from 2D planes close to the cell's center of mass does not accurately reflect the complexity of real world applications. Therefore, we challenged SHAPR with the task of predicting cell nuclei shapes from confocal z-stacks containing fluorescence counterstained nuclei from human-induced pluripotent stem cells (iPSCs) cultured in a spheroidal aggregate. To generate the ground truth data, 887 cell nuclei from six iPSC-derived aggregates were manually segmented in 3D (Figure 2A). SHAPR was provided with one 2D image slice taken at an aggregate depth of 22 μm and it is the corresponding segmentation as input (Figure 2B). Nuclei were thus cut at random heights, leading to a variety of segmented areas, markedly complicating the prediction of 3D shapes (Figure 2C). Following this, we trained five SHAPR models during cross-validation. Predictions were compared to cylindrical and ellipsoid fits, as described above. Again, the relative volume error was significantly lower for SHAPR (Figure 2D; $\text{Volume}_{\text{SHAPR}} = 33 \pm 41$ vs. $\text{Volume}_{\text{Cylindrical}} = 44 \pm 25$, $p = 9.2 \times 10^{-36}$, and $\text{Volume}_{\text{Ellipsoid}} = 62 \pm 29$, $p = 8.7 \times 10^{-86}$, n = 887, paired Wilcoxon signed-rank test) compared to the naïve models. More importantly, SHAPR predictions were also closer to the true nuclei shapes in terms of volume and surface area compared with cylindrical and ellipsoid model predictions (Figure 2D). To determine how much information our model had learned about nuclear shape, we compared the 2D segmentation area with the volume of the ground truth, the SHAPR predictions, and the cylindrical and ellipsoid fits (Figure 2E). As expected, the cylindrical and ellipsoid models were simply extrapolating area to volume monotonically, whereas the ground truth suggested a more complex relationship. SHAPR was able to learn that small segmentation areas (<200 pixels) do not emerge from minuscule nuclei but from slices at the nuclear edge. Notably, our model could only obtain high intersection over union scores for slices close to the center of mass, in contrast to volume and surface error (Figure 2F). This suggests that it can predict volume and surface correctly, but not if a nucleus is cut in its upper or the lower half.

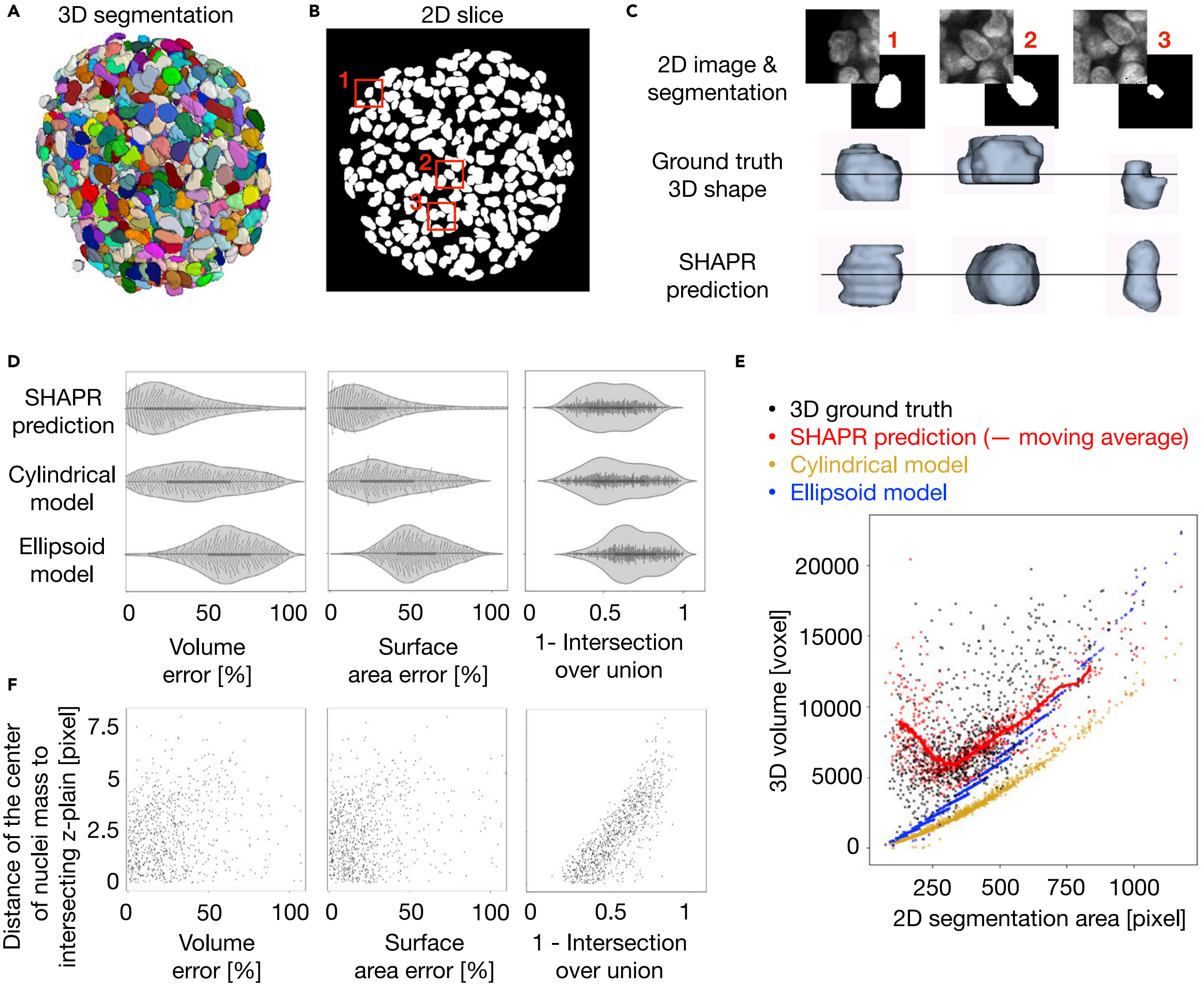


Figure 2. SHAPR learns fundamental 3D shape properties of human-induced pluripotent stem cell (iPSC) nuclei from a single 2D slice

(A) Representative image of a segmented human-iPSC derived 3D cell culture with fluorescently stained nuclei. To generate ground truth data, six 3D cell cultures were manually segmented.

(B) 2D nuclei segmentation from a single slice at 22μm depth.

(C) 2D segmentation areas and fluorescent image intensities varied considerably with the position of the intersecting slice.

(D) SHAPR predictions outperform cylindrical and ellipsoid models in terms of volume, surface area error and intersection over union.

(E) Although the cylindrical and ellipsoid models are only able to extrapolate volumes in a naïve manner, SHAPR learned complex, non-linear relationships between the 2D segmentation area and the 3D volume of a nucleus.

(F) Although the intersection over union decreased with distance to the nucleus center of mass, volume, and surface predictions were unaffected.

DISCUSSION

SHAPR is able to solve the ambiguous inverse problem of predicting 3D shapes of single cells and nuclei from 2D images, shown on two different datasets. Although Wu et al.'s (Wu et al., 2019) approach can predict the axial position of single fluorophores from a 2D image, and Ounkomol et al. (Ounkomol et al., (2018) have shown that fluorescent markers can be accurately predicted from 3D brightfield images, which is an image to image translation task, SHAPR performs a spatial reasoning task, considering contextual information to reconstruct the shape of cells and nuclei. SHAPR is however not able to reconstruct information that is inaccessible or coming from far away from the imaged plane. Reconstructing 3D shapes is an ambiguous inverse problem because it is not clear if voxels would lie over or under the imaged 2D plane that SHAPR was provided with. Furthermore, each 2D image permits numerous 3D reconstructions, similar to how a shadow alone does not permit precise conclusions about the corresponding shape. Position-invariant metrics such as the volume error or surface area error should thus be used as alternative evaluation metrics beyond the IoU. Although there are some outliers with a high reconstruction error, SHAPRs was generally able to retrieve real-world 3D information and outperformed naïve 3D shape fitting models on both datasets. Furthermore we have shown that classification accuracy for red blood cells was significantly improved using the features provided by SHAPR, as opposed to the features extracted from the 2D images. Predicting 3D shapes from 2D images thus offers a simple way to reduce imaging time and data storage while retaining morphological details. A trained SHAPR model could thus enable efficient predictions of single-cell volume distributions and density, e.g. to screen organoids and identify outlier events. In combination with in-silico staining approaches (Ounkomol et al., 2018; Christiansen et al., 2018; Rivenson et al., 2019), SHAPR could be used for label-free single cell classification, e.g., for diagnostic purposes in computational pathology (O'Connor et al., 2020,2021; Anand et al., 2017). We are curious to further explore SHAPR's potential on phenotypic data, where cell morphologies may be subject to change and a model trained on wildtype data might have difficulties to generalize. As a general framework SHARP is not limited to confocal fluorescence images, and it will be particularly interesting to integrate different image modalities in the future. Also, utilizing multiple 2D slices as an input for SHAPR could prove beneficial, as well as using more informative losses that e.g. incorporate topological information (Horn et al., 2021). Our well documented open source package of SHAPR, available on GitHub, allows for easy extension of the data loader and adaptations of the loss function used. Going beyond single-cell shape prediction, our approach may be extended to other biological structures, including organelles and proteins, and may increase the efficiency of biomedical imaging in multiple domains.

Limitations of the study

SHAPR may predict cell shapes with a high relative error to the ground truth for some samples. The binary cross entropy and dice loss used during training are not able to regularize contextual information, but only geometrical information. We explore a possible solution to this by incorporating a topology based loss function (Waibel et al., 2022).

STAR★METHODS

Detailed methods are provided in the online version of this paper and include the following:

- KEY RESOURCES TABLE
- RESOURCE AVAILABILITY
  - Lead contact
  - Data and code availability
- METHOD DETAILS
  - SHAPR
  - Training parameters
  - Evaluation metrics
  - Feature extraction
  - Feature-based classification
  - Datasets

SUPPLEMENTAL INFORMATION

Supplemental information can be found online at https://doi.org/10.1016/j.isci.2022.105298.

ACKNOWLEDGMENTS

We thank Mohammad Mirkazemi and Ron Fechtner for reviewing our code. We thank Sophia Wagner, Tingying Peng, Sayedali Shetab Boushehri, and Matthias Hehr for discussions and for contributing their ideas. We thank Marius Bäuerle and Valerio Lupperger for their feedback on the figures and manuscript.

Funding: CM has received funding from the European Research Council (ERC) under the European Union's Horizon 2020 research and innovation programme (Grant agreement No. 866411).

AUTHOR CONTRIBUTIONS

D.W. and N.K. implemented code and conducted experiments. D.W., N.K., and C.M. wrote the manuscript with A.S., S.A. and M.M. D.W. created figures and the main storyline with C.M., S.A., and M.M. provided the 3D cell culture dataset and ideas. C.M. supervised the study. All authors have read and approved the manuscript.

DECLARATION OF INTERESTS

The author(s) declare no competing interests.

Received: January 10, 2022
Revised: May 4, 2022
Accepted: September 28, 2022
Published: November 18, 2022

REFERENCES

Abadi, M., Agarwal, A., Paul, B., Brevdo, E., Chen, Z., Craig, C., Corrado, G.S., et al. (2016). TensorFlow: large-scale machine learning on heterogeneous distributed systems. Preprint at arXiv. http://arxiv.org/abs/1603.04467.

Anand, A., Moon, I., and Javidi, B. (2017). Automated disease identification with 3-D optical imaging: a medical diagnostic tool. Proc. IEEE *105*, 924–946. https://doi.org/10.1109/jproc.2016.2636238.

Basic Installation — Trimesh 3.9.24 Documentation, n.d. https://trimsh.org/.

Boulogne, F., Warner, J.D., Nunez-Iglesias, J., Boulogne, F., Warner, J.D., Yager, N., Gouillart, E., and Yu, T. (2014). Scikit-image: image processing in Python. PeerJ *2*, 453.

Breiman, L. (2001). Random forests. Mach. Learn. *45*, 5–32.

Chang, A.X., Funkhouser, T., Guibas, L., Hanrahan, P., Huang, Q., Li, Z., Savarese, S., et al. (2015). ShapeNet: an information-rich 3D model repository. Preprint at arXiv. https://doi.org/10.48550/arXiv.1512.03012.

Chen, M., and Boyle, F.J. (2017). An enhanced spring-particle model for red blood cell structural mechanics: application to the stomatocyte-discocyte-echinocyte transformation. J. Biomech. Eng. *139*. https://doi.org/10.1115/1.4037590.

Chollet, F., et al. (2015). Keras. https://keras.io.

Choy, C.B., Xu, D., Gwak, J., Chen, K., and Savarese, S. (2016). 3D-R2N2: a unified approach for single and multi-view 3D object reconstruction. In Computer Vision – ECCV 2016, pp. 628–644. Springer International Publishing.

Christiansen, E.M., Yang, S.J., Ando, D.M., Javaherian, A., Skibinski, G., Lipnick, S., Mount, E., O'Neil, A., Shah, K., Lee, A.K., et al. (2018). In silico labeling: predicting fluorescent labels in unlabeled images. Cell *173*, 792–803.e19.

Fan, H., Su, H., and Guibas, L. (2017). A point set generation network for 3D object reconstruction from a single image. 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR). https://doi.org/10.1109/cvpr.2017.264.

Gkioxari, G., Malik, J., and Johnson, J. (2019). Mesh R-Cnn." in Proceedings of the IEEE/CVF International Conference on Computer Vision, pp. 9785–9795.

Horn, M., De Brouwer, E., Moor, M., Moreau, Y., Rieck, B., and Borgwardt, K. (2021). Topological graph neural networks. Preprint at arXiv. https://doi.org/10.48550/arXiv.2102.07835.

Javidi B., Carnicer A., Anand A., and Barbastathis G. n.d. "Roadmap on Digital Holography." Optics. https://opg.optica.org/abstract.cfm?uri=oe-29-22-35078.

Kingma, D.P., and Jimmy, B. (2014). Adam: a method for stochastic optimization. Preprint at arXiv. https://doi.org/10.48550/arXiv.1412.6980.

Lewiner, T., Lopes, H., Vieira, A.W., and Tavares, G. (2003). Efficient implementation of marching cubes' cases with topological guarantees. J. Graph. Tool. *8*, 1–15.

Martínez-Corral, M., and Javidi, B. (2018). Fundamentals of 3D imaging and displays: a tutorial on integral imaging, light-field, and plenoptic systems. Adv. Opt. Photonics *10*, 512.

Moon, I., Anand, A., Cruz, M., and Javidi, B. (2013). Identification of malaria-infected red blood cells via digital shearing interferometry and statistical inference. IEEE Photon. J. *5*, 6900207.

O'Connor, T., Anand, A., Andemariam, B., and Javidi, B. (2020). Deep learning-based cell identification and disease diagnosis using spatio-temporal cellular dynamics in compact digital holographic microscopy. Biomed. Opt Express *11*, 4491–4508.

O'Connor, T., Shen, J.-B., Liang, B.T., and Javidi, B. (2021). Digital holographic deep learning of red blood cells for field-portable, rapid COVID-19 screening. Opt. Lett. *46*, 2344–2347.

Ounkomol, C., Seshamani, S., Maleckar, M.M., Collman, F., and Johnson, G.R. (2018). Label-free prediction of three-dimensional fluorescence images from transmitted-light microscopy. Nat. Methods *15*, 917–920. https://doi.org/10.1038/s41592-018-0111-2.

Rivenson, Y., Liu, T., Wei, Z., Zhang, Y., de Haan, K., and Ozcan, A. (2019). PhaseStain: the digital staining of label-free quantitative phase microscopy images using deep learning. Light Sci. Appl. https://doi.org/10.1038/s41377-019-0129-y.

Simionato, G., Hinkelmann, K., Chachanidze, R., Bianchi, P., Fermo, E., van Wijk, R., Leonetti, M., Wagner, C., Kaestner, L., and Quint, S. (2021). Red blood cell phenotyping from 3D confocal images using artificial neural networks. PLoS Comput. Biol. *17*, e1008934.

Sun, X., Wu, J., Zhang, X., Zhang, Z., Zhang, C., Xue, T., Tenenbaum, J.B., and Freeman, W.T. (2018). Pix3d: dataset and methods for single-image 3d shape modeling. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 2974–2983.

Sung, Y., Choi, W., Fang-Yen, C., Badizadegan, K., Dasari, R.R., and Feld, M.S. (2009). Optical diffraction tomography for high resolution live cell imaging. Opt Express *17*, 266–277.

Varoquaux, G., Buitinck, L., Louppe, G., Grisel, O., Pedregosa, F., and Mueller, A. (2015). Scikit-learn. GetMobile: Mobile Comp. and Comm. *19*, 29–33.

Waibel, D.J.E., Atwell, S., Meier, M., Marr, C., and Rieck, B. (2022). Capturing Shape Information with Multi-scale Topological Loss Terms for 3D Reconstruction. In Lecture Notes in Computer Science, *13434*, L. Wang, Q. Dou, P.T. Fletcher, S. Speidel, and S. Li, eds (Springer). https://link.springer.com/chapter/10.1007/978-3-031-16440-8_15.

Wang, N., Zhang, Y., Li, Z., Fu, Y., Liu, W., and Jiang, Y.-G. (2018). Pixel2mesh: generating 3d mesh models from single rgb images. In Proceedings of the European Conference on Computer Vision (ECCV), pp. 52–67.

Weigert, M., Schmidt, U., Boothe, T., Müller, A., Dibrov, A., Jain, A., Wilhelm, B., Schmidt, D., Broaddus, C., Culley, S., et al. (2018). Content-aware image restoration: pushing the limits of fluorescence microscopy. Nat. Methods *15*, 1090–1097.

Wu, Y., Rivenson, Y., Wang, H., Luo, Y., Ben-David, E., Bentolila, L.A., Pritz, C., and Ozcan, A. (2019). Three-dimensional virtual refocusing of fluorescence microscopy images using deep learning. Nat. Methods *16*, 1323–1331. https://doi.org/10.1038/s41592-019-0622-5.

Xiang, Y., Mottaghi, R., and Savarese, S. (2014). Beyond PASCAL: a benchmark for 3D object detection in the wild. In IEEE Winter Conference on Applications of Computer Vision, pp. 75–82.

STAR★METHODS

KEY RESOURCES TABLE

REAGENT or RESOURCE	SOURCE	IDENTIFIER
Deposited data		
Red blood cell dataset	(Simionato et al., 2021)	https://doi.org/10.5281/zenodo.7031924
Nuclei Dataset	This paper	https://doi.org/10.5281/zenodo.7031924
Software and algorithms		
SHAPR	This paper	https://github.com/marrlab/SHAPR

RESOURCE AVAILABILITY

Lead contact

Further information and requests for resources and reagents should be directed to and will be fulfilled by the lead contact, Carsten Marr (carsten.marr@helmholtz-muenchen.de).

Data and code availability

- The datasets are available at: https://doi.org/10.5281/zenodo.7031924.
- The code of SHAPR is available as a well documented, pip installable package together with commented analysis scripts at https://github.com/marrlab/SHAPR.

METHOD DETAILS

SHAPR

Our SHApe PRediction algorithm SH APR *S* consists of an encoder and a decoder with parameters θ (see Figure S1A) that transforms an 2D input $i \in I$, which is a 2D fluorescent image and a corresponding binary mask (see Figure 1A), to a binary 3D output *p*:

$$p = S(i;\theta).$$

A discriminator *D* with parameters τ tries to distinguish if a 3D shape *x* comes from SHAPR or from real data:

$$l = D(x;\tau).$$

Parameters θ and τ are learned during training when the objective function $\mathcal{L}$ is minimized:

$$\mathcal{L} = \mathcal{L}_{rec} + \alpha \left(\mathcal{L}_{adv} + \mathcal{L}_{dis}\right).$$

Here, $\mathcal{L}_{rec}$ is the reconstruction loss, $\mathcal{L}_{adv}$ is the adversarial loss, $\mathcal{L}_{dis}$ is the discriminator loss and α regulates the impact of adversarial and discriminator loss during training. The reconstruction loss tries to match the generated 3D output *p* with 3D ground truth *y* and is defined as:

$$\mathcal{L}_{rec}(\theta) = d(p,y) + b(p,y)$$

where $d(.,.)$ and $b\,(.,.)$ are Dice loss and binary cross entropy loss, respectively (see Figure S1A). The adversarial loss tries to match the distribution of generated shapes with the dataset ground truth:

$$\mathcal{L}_{adv}(\theta) = \mathbb{E}_{i\,\in\,I} \, log\,(1 - \mathrm{D}(\mathrm{S}(\mathrm{i};\theta),\tau).$$

The discriminator loss is defined by:

$$\mathcal{L}_{dis}(\tau) = \mathbb{E}_{y\,\in\,Y} \, log\, D(y;\tau) + \mathbb{E}_{i\,\in\,I} \, log(1 - D(S(i;\theta),\tau).$$

In the following implementation details are explained in further details: The encoder is built of three blocks (see Figure S1A). Each block contains two 3D convolutional layers with a kernel size of (1,3,3), followed by a batch normalization, a dropout, and a max pooling layer to downsample convolutions. The last activation function of the encoder is a sigmoid. The decoder consists of seven convolutional blocks and each of them contains two 3D convolutional layers, followed by batch normalization, a dropout, and a 3D transpose convolutional layer for upsampling. We upsample the z-dimension seven times and the x-y dimensions 3 times in an alternating fashion. The discriminator D consists of five convolutional layers with a kernel size of (3,3,3) followed by an average-pooling in each dimension and two dense layers, one with 128 and one with 1 unit, followed by a sigmoid activation function, which outputs a binary label for each 3D input shape. The regularization parameter α is a step function starting with 0 so the model is trained using the reconstruction loss alone. After 30 epochs or if the validation loss has not improved for 10 epochs, α switches to 1. From then on, SHAPR is training in an adversarial fashion. Model was implemented using Tensorflow and Keras (Abadi et al., 2016; Chollet et al., 2015).

Training parameters

Five independent models were trained on both datasets in a round-robin fashion, so each input image was contained in the test set exactly once. For each model 20% of the dataset was used as a held out test set. 20% of the remaining data was used as a validation set during training. The remaining 60% was used to optimize SHAPRs model weights during training. SHAPRs hyperparameters, such as the learning rate and number of model weights have been fixed before training. Adam optimizer (Kingma and Jimmy, 2014) with an initial learning rate of $1*10^{-3}$, beta1 of 0.9, and beta2 of 0.999 were used. For data augmentation, training data was randomly flipped horizontally and vertically and rotated with a chance of 33% for each augmentation to be applied on each data point.

To obtain a binary image all SHAPR predictions are thresholded at 126, as their pixel values range from 0 to 255.

Evaluation metrics

For comparison with different models, five metrics are used: relative voxel error, relative volume error, relative surface error, relative surface roughness error and intersection over union (IoU). With Y being the ground truth and P the prediction, these are defined as:

$$IoU(Y,R) = \frac{Y \cap P}{Y \cup P}$$

$$Relative\ voxel\ error\ (Y,P) = \frac{1}{NMK}\sum_{x=1}^{N}\sum_{y=1}^{M}\sum_{z=1}^{K}\left|\frac{Y_{xyz} - P_{xyz}}{Y_{xyz}}\right|$$

$$Relative\ volume\ error\ (Y,P) = \frac{volume\ (Y) - volume\ (P)}{volume\ (Y)}$$

$$Relative\ surface\ error\ (Y,P) = \frac{surface\ (Y) - surface\ (P)}{surface\ (Y)}$$

$$Relative\ surface\ roughness\ error\ (Y,P) = \frac{surface\ roughness\ (Y) - surface\ roughness\ (P)}{surface\ roughness\ (Y)}$$

where N, M, and K are the bounding box sizes and *volume*(.) and *surface*(.) yield the volume by counting non-zero voxels:

$Volume\ (P) = \sum_{x=1}^{N}\sum_{y=1}^{M}\sum_{z=1}^{K}\mathbb{1}(P_{xyz} > 0)$, where $\mathbb{1}$ is the indicator function.

and surface area by counting all voxels on the surface of a given 3D binary shape:

$Surface\ (P) = \sum_{x=1}^{N}\sum_{y=1}^{M}\sum_{z=1}^{K}\mathbb{1}(P_{xyz} = \partial P)$, where ∂P is defined as the surface of P.

The function *surface roughness*(.) is defined as:

$$Surface\ roughness\ (P) = \sum_{x=1}^{N}\sum_{y=1}^{M}\sum_{z=1}^{K}\left|P_{xyz} - P_{xyz}^{gaussian}\right|$$

with:

$$P_{xyz}^{gaussian} = \frac{1}{\left(\sqrt{2\pi}\,\sigma\right)^3} exp\left(-\frac{x^2+y^2+z^2}{2\,\sigma^2}\right), \quad \forall x,y,z \in P$$

Feature extraction

We extract 126 features from each 3D shape, comprising volume, surface, shape index, roughness, convexity, and gabor features with NumPy and the Skimage toolbox (Table S3) (Boulogne et al., 2014). Eleven features are derived by describing the object with a mesh consisting of faces and vertices. The mesh is calculated using the marching cubes algorithm in python's Skimage toolbox (Lewiner et al., 2003; Boulogne et al., 2014), resulting in faces and vertices. The surface, 19 mesh principals, the first nine mesh inertia eigenvalues were calculated using trimesh ("Basic Installation — Trimesh 3.9.24 Documentation" n.d.). The moments of inertia represent the spatial distribution of mass in a rigid 3D shape. This depends on the 3D shapes mass, size, and shape. The moment of inertia is calculated as the angular momentum divided by the angular velocity around a principal axis. We also calculate the objects' moments up to the third order, correlation and dissimilarity of the gray level co-occurrence matrices, and one gabor feature for each z-slice, resulting in 64 gabor features all using the Skimage toolbox (Boulogne et al., 2014).

From 2D segmentations, we extract 9 features. These are the mean pixel value, area, boundary length, boundary roughness, convexity, two moments, and three gabor features and 5 features from the 2D microscopy images (see Table S3).

Feature-based classification

To establish the baseline for the feature-based classification, we extract 9 morphological features (mean pixel value, area, boundary length, boundary roughness, convexity, two moments, and three gabor features) and 5 features from the 2D microscopy images that have been multiplied with the respective segmentation mask to reduce noise (mean and standard deviation of the pixel intensity, one gabor feature, and the correlation and dissimilarity of the gray level co-occurrence matrices). In total, we extract 15 features from each of the paired 2D images and 2D segmentations. We compared random forest, decision tree, K-nearest-neighbors, linear discriminant analysis, naïve Bayes, and support vector machine classifiers using the Sklearn toolbox (Varoquaux et al., 2015). The random forest classifier with 1000 estimators performed best.

Prior to the classification we oversample the training data to compensate for class imbalance, and normalize all features by subtracting their mean and dividing by their standard deviation. Prior to the tenfold cross-validation classification, we trained one random forest model to investigate feature importance. We found that accuracy increases if we remove features with an importance lower than 0.005 (see Figure S1C). We do not only achieve an overall higher F1 score using ShapeAEs predictions, but increase the number of true positives for four of the six classes, while for two classes we obtain the same scores (Figure S1A).

Datasets

Red blood cells

We use 825 publicly available 3D images of red blood cells (Simionato et al., 2021) of size (64, 64, 64) voxels, each assigned to one of the following six classes: SDE shapes, cell cluster, multilobate, keratocyte, knizocyte, and acanthocyte. Spherocytes, stomatocytes, discocytes, and echinocytes are combined into the SDE shapes class, which is characterized by the stomatocyte–discocyte–echinocyte transformation (Chen and Boyle, 2017). The other classes' shapes occur in samples from patients with blood disorders or other pathologies.

The number of cells in each class were: 602 for SDE shapes (93 spherocytes, 41 stomatocytes, 176 discocytes, and 292 echinocytes), 69 for cell clusters, 12 multilobates, 31 keratocytes, 23 knizocytes, and 88 acanthocytes. Red blood cells were drawn from 10 healthy donors and 10 patients with hereditary spherocytosis via finger-prick blood sampling and then fixed. Thereafter the cells were imaged with a confocal microscope and then manually classified. We extracted the 2D image from the central slide of each 3D image and segmented it by thresholding. Also, the 3D ground truth was obtained by thresholding.

Human pluripotent stem cells derived 3D cultures

Six human induced pluripotent stem cell-derived 3D cultures were imaged with a Zeiss LSM 880 Airyscan inverted confocal microscope with DAPI as a nuclear counterstain. Full 3D stacks were acquired using a 20× objective with a resolution of 0.25µm/pixel and a distance between slices of 1µm. We rescaled the images to 0.5µm/pixel in x-y dimension. From the center z-slice of each 3D cell culture, we manually segment and isolate all nuclei in 2D and the corresponding 3D nuclei, resulting in a dataset of 887 paired 2D/3D single-cell shapes of shapes (64,64) pixel and (64, 64, 64) pixel. We interpolated the single 3D nuclei to an isotropic spacing of 0.5µm/pixel. While for the red blood cell dataset we could expect to cut each single cell roughly in its middle, in this dataset the nuclei are cut at any possible z-position.

A.3. Capturing Shape Information with Multi-Scale Topological Loss Terms for 3D Reconstruction

Dominik J. E. Waibel, Scott Atwell, Matthias Meier, Carsten Marr, and Bastian Rieck

This article was published in International Conference on Medical Image Computing and Computer-Assisted Intervention (MICCAI) following peer review.

A.4. Label-free imaging of 3D pluripotent stem cell differentiation dynamics on chip

Scott Atwell*, Dominik J. E. Waibel*, Sayedali Shetab Boushehri, Carsten Marr, and Matthias Meier
* contributed equally

Label-free imaging of 3D pluripotent stem cell differentiation dynamics on chip

Scott Atwell[1*], Dominik J. E. Waibel[2,3,*], Sayedali Shetab Boushehri[2,4,5], Carsten Marr[2], Matthias Meier[1,6]

[1]Helmholtz Pioneer Campus, Helmholtz Munich, Munich, Germany
[2]Institute of AI for Health, Helmholtz Munich, Neuherberg, Germany
[3]TUM School of Life Sciences, Technical University of Munich, Weihenstephan, Germany
[4]Department of Mathematics, Technical University of Munich, Munich, Germany
[5]Roche Innovation Center Munich, Roche Diagnostics GmbH, Penzberg, Germany
[6]TUM School of Medicine, Technical University of Munich, Munich, Germany
*Contributed equally
Corresponding authors: Dr. Carsten Marr and Dr. Matthias Meier
Email: carsten.marr@helmholtz-muenchen.de, matthias.meier@helmholtz-muenchen.de

Abstract

The dynamic chemical and architectural microenvironments of 3D stem cell cultures can be controlled by integration into a microfluidic chip. Massive parallelized 3D stem cell cultures for engineering *in vitro* human cell types require new imaging methods with high time and spatial resolution to fully exploit technological advances in cell culture. Here, we introduce a label-free deep learning method called Bright2Nuc to predict *in silico* nuclear staining in 3D from bright-field images obtained using traditional confocal microscopy. Bright2Nuc was trained and applied to several hundred 3D human induced pluripotent stem cell cultures differentiating towards definitive endoderm on a microfluidic platform. Combined with existing image analysis tools, Bright2Nuc segmented individual nuclei from bright-field images, quantified their morphological properties, predicted stem cell differentiation state, and tracked the cells over time. Our methods are available in an open-source pipeline that enables researchers to upscale 3D cell phenotyping in stem cell culture.

Introduction

New cell culture technologies for pluripotent stem cells are central to enabling *in vitro* disease models, regenerative therapies, and animal replacing drug screens [1]. 3D culture techniques have become commonly used in differentiation trials of stem cells towards defined cell types, such as beta cells [2], hepatocytes [3], and intestinal epithelial cells [4]. Although 3D cultures can mimic the architecture of the tissue niche, promoting cell-to-cell and cell-to-matrix interactions, the chemical microenvironment requires additional technological tools. Microfluidic chip technologies can fill this gap by automating fluid programs to test complex differentiation protocols in parallelized stem cell cultures [5]. Further, microfluidics offers miniaturized solutions for unifying the shape and size of 3D stem cell cultures, but also for positioning the cell cultures to simplify high-content data acquisition. With the progression of cell culture technologies, analytical methods to phenotype 3D cell cultures under higher throughput conditions have become difficult because cells in 3D adopt high-density compact configurations, and cell morphologies are more heterogeneous and less recognizable than in 2D cultures. Thus, tracking and phenotyping massively parallelized 3D cultures at the single-cell level presents the challenge of not only developing image acquisition methods allowing to resolve the 3D cell cultures in high time and spatial resolution, but also powerful computational methods to handle the increasing amount of data being generated [6].

Owing to the widely accessible equipment, confocal fluorescence is the gold standard for obtaining single-cell resolutions in optical microscopy. Real-time imaging of cell types, functions, or cell states is hampered by the need for fluorescent reporter cell lines, which often require lengthy processes to engineer and validate [7]. Furthermore, fluorescent reporters may affect biochemical phenomena or cell types of interest and induce cytotoxic stress under prolonged imaging times. Label-free microscopy, based on deep learning image processing and analysis, has emerged as an alternative to fluorescent reporters. For example, *in silico* staining can be used to predict fluorescent markers from the bright-field images of various tissue types [8–11]. Once trained, deep-learning models are fast and consistent in their predictions. For high-content imaging in screening studies, label-free microscopy can dramatically reduce the image acquisition time by inferring multiple fluorescent *in silico* stains from single images without being limited by spectral cross-talk. Most of the previous approaches have focused on high-resolution imaging of 2D cell cultures or tissue sections with high numerical apertures. Higher-throughput imaging results in a trade-off between acquisition time, resolution, and phototoxicity. In particular, a higher resolution is only achieved with liquid immersion objectives, which are cumbersome for high-content data acquisition. Further, though not unprecedented [12], applications of *in silico* staining to live 3D cell cultures are rare. Finally, no previous approach has addressed the problem of phenotyping stem cells in a 3D environment from bright-field images.

Herein, we present a microfluidic platform for complex fluid programming in a multiplexed high-throughput manner to enable the differentiation of human induced pluripotent stem cells (hiPSC) in 3D, while simultaneously facilitating live imaging. We focused on acquiring imaging data on 3D cell cultures with a relatively low-resolution and low-numerical-aperture air-immersion 20X objective through confocal microscopy to increase data acquisition throughput. We introduced a deep learning-based method, named Bright2Nuc, to predict nuclear fluorescence within on-chip 3D cell cultures from low-magnification bright-field images. *In silico* staining images were used to infer the differentiation state of human stem cells and cell dynamics within on-chip cultivated 3D cell cultures undergoing endodermal differentiation. For reproducibility and applicability, the Bright2Nuc deep learning method and associated data were published using open-source software.

Results

Imaging of 3D human stem cell cultures on chip

We developed a microfluidic large-scale integration chip platform to automate the formation, culture, and differentiation of 128 3D cell cultures under 32 independent chemical conditions (**Fig. 1a**). Cells were seeded as a single-cell suspension in each chamber, and after 4 h, 3D cell cultures were formed by self-aggregation (**Fig. 1b**). A chamber height of 50 µm constrained the 3D cell cultures between the glass substrate and PDMS channel top, ensuring imaging from top to bottom (**Fig. 1c**). Together with the position stability of the cell culture compartments, the chip enabled high-content time-resolved imaging of reproducibly homogeneous (**Supplementary Fig. 1a**) live 3D cell cultures with temporal control of chemical conditions.

Within the first 24 h after the start of DE differentiation, 3D cell cultures lost an average 42.6 ± 17.1% (mean ± SD, averaged over N_r = 4 biological replicates with a total of N = 426 cultures) of their surface area in the xy-plane due to cell disaggregation (**Supplementary Fig. 1b and d**). After this initial cell loss, most of the cell cultures recovered and grew until the end of differentiation (**Supplementary Fig. 1c**). The individual growth behavior of 3D cell cultures varied depending on the initial cell loss, but TF expression during differentiation was comparable for all cell cultures, where hiPSCs lost the expression of the pluripotency marker

OCT4 within the first 48 h and gained the expression of the DE-specific markers, FOXA2 and SOX17 (**Fig. 1d**).

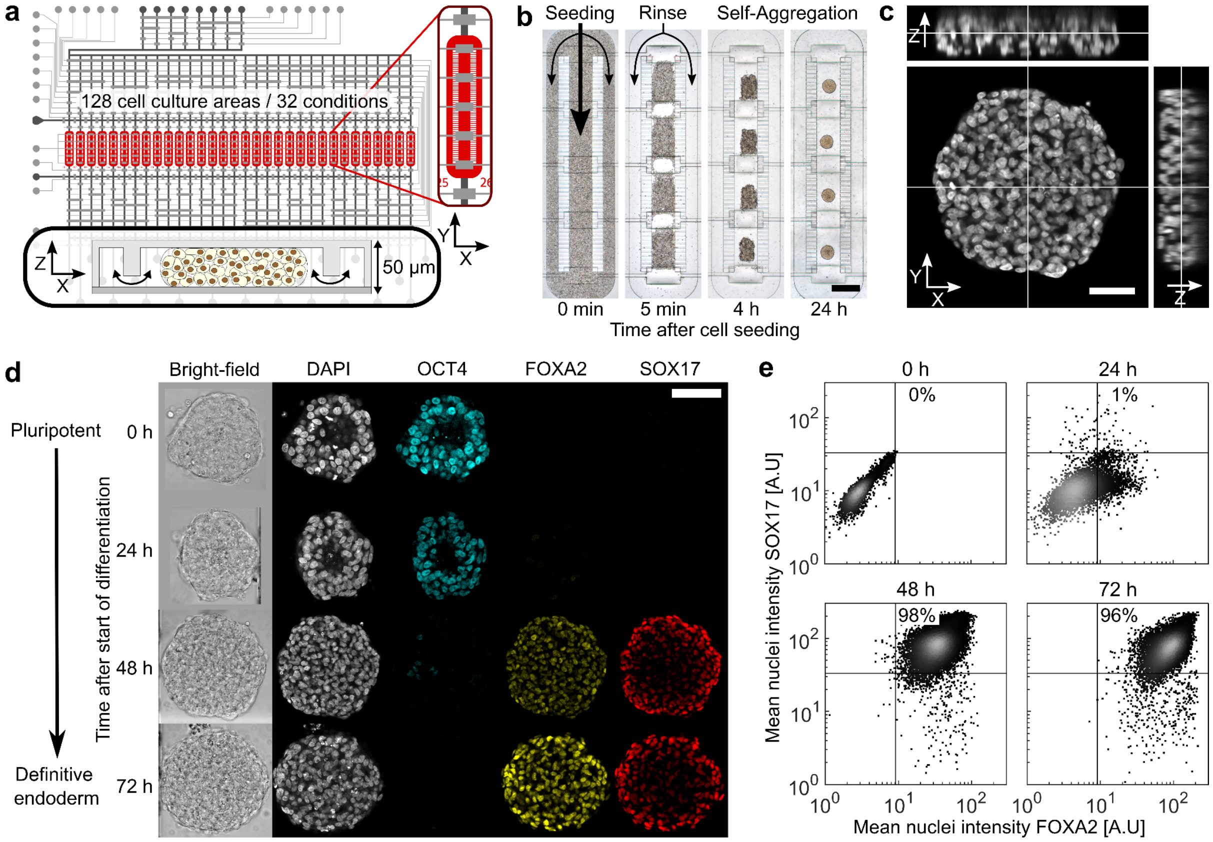


Figure 1 | Microfluidic large-scale integration chip platform chip enables the multiplexed differentiation and imaging of 3D human induced pluripotent stem cell cultures.
a, Scheme of microfluidic large-scale integration chip layout. The black and gray lines denote the flow and control microchannels of the double-layered PDMS chip, respectively. The inset on the right shows a magnified view of one cell culture chamber (red). The bottom inset shows the cross-section through a filled cell culture area. **b,** On-chip 3D cell culture formation process: i) seeding of single-cell solution, ii) separation of cell culture areas upon actuation of pneumatic membrane valves, followed by cell rinsing, and iii) self-aggregation of cells in the confined cell culture volume after 4 h and 24 h. Scale bar: 500 µm. **c,** Orthogonal views of an on-chip 3D cell culture stained with DAPI. Images were taken with a 0.25 µm/px xy-resolution and 1 µm/px z-resolution. Scale bar: 50 µm. **d,** Immunofluorescence confocal images of 3D cell cultures fixed along the differentiation trajectory from the pluripotency (OCT4 positive) to the definitive endoderm (FOXA2 and SOX17 double-positive) cell stage. Scale bar: 100 µm. **e,** Quantitative image analysis of cell type marker expression along the differentiation trajectory within all cells of 3D cell cultures from a single chip run. Each data point represents the mean fluorescence intensity of individual nuclei segmented in 3D; the indicated percentages represent the fraction of FOXA2/SOX17 double-positive nuclei (nuclei considered: n_{0h} = 5501, n_{24h} = 8700, n_{48h} = 22587, n_{72h} = 14936) from multiple 3D cell cultures (N_{0h} = 12, N_{24h} = 25, N_{48h} = 24, N_{72h} = 30).

Furthermore, on-chip differentiation resulted in a nearly homogenous cell population with a yield as high as 96 ± 3% FOXA2/SOX17 double-expressing cells after 72 h (**Fig. 1e**, mean ±

SD, n = 14936 nuclei in N = 20 cultures). However, experimental yields varied with an average of 90 ± 6% (mean ± SD, N_r = 4) of double-expressing cells at the end of differentiation, as previously observed in the standard cell well plate and chip culture formats [15].

We applied our platform to definitive endoderm (DE) differentiation, a critical first step for differentiating liver, gut, pancreas, lungs, trachea, and thyroid cell types. hiPSC-derived 3D cultures could be differentiated on-chip towards DE by activating the TGF-β/nodal and WNT signaling pathways with activin A and CHIR-99021, respectively. Over a period of 3 days, the two chemical components were controlled in time and concentration using fluid programming (**Methods**). Upon fixation of subsets of 3D cell cultures with NHS-ester every 24h, a differentiation trajectory was established (**Supplementary Fig. 2**). All cell cultures were immunostained on the chip at the end of differentiation for the pluripotency marker octamer-binding transcription factor 4 (OCT4) and the two DE-specific transcription factors (TFs) forkhead box A2 (FOXA2) and SRY-box 17 (SOX17). Bright-field and immunofluorescence (IF) images were recorded by standard confocal microscopy with a xy-resolution of 0.25 µm/px and a z-resolution of 1 µm/plane. Quantitative fluorescence signals of the TFs were extracted per nucleus for all 3D cell cultures by segmenting the 3D DAPI signal using a re-trained StarDist model [13,14]. All IF images were corrected for the signal decrease in the z-direction caused by increasing light scattering and for the signal decrease in the xy-direction caused by gradually decreasing penetration of labeled antibodies (**Methods**).

Label-free prediction of nuclei in 3D cell cultures

The full high-content imaging of hiSPC-derived 3D cell cultures with standard confocal microscopy with four fluorescence channels is laborious and time-consuming, requiring approximately 48 h for IF staining and 48 h of imaging. To cope with the chip throughput and characterize live cell cultures, we sought to predict the nuclear fluorescence of cells within whole 3D cell cultures along the DE differentiation trajectory from low-magnification bright-field images. For this purpose, we developed Bright2Nuc, a U-Net-based deep neural network (**Fig. 2a and b**). Bright2Nuc was trained on 255 paired bright-field stacks (one image below and one above a central slice) and fluorescence nucleus images of 181 live and 74 fixed 3D cell cultures by minimizing the mean-squared pixel error between the paired predicted and true IF nucleus images (see **Fig. 2a and Methods**). While images from fixed 3D cell cultures were acquired along the DE trajectory as described previously (**Fig. 1c**), images from live 3D cell cultures were obtained in the pluripotent state using a

SOX2-T2A-tdTomato fluorescence reporter hiPSC line (**Supplementary Fig. 3**). Using the local 3D information from three neighboring z slices as input (**Fig. 2b**), the trained Bright2Nuc model accurately predicted nucleus images in an independent test set containing 85 cell cultures (26 fixed, 59 live) from the same experiments completely excluded from training (**Fig. 2c**).

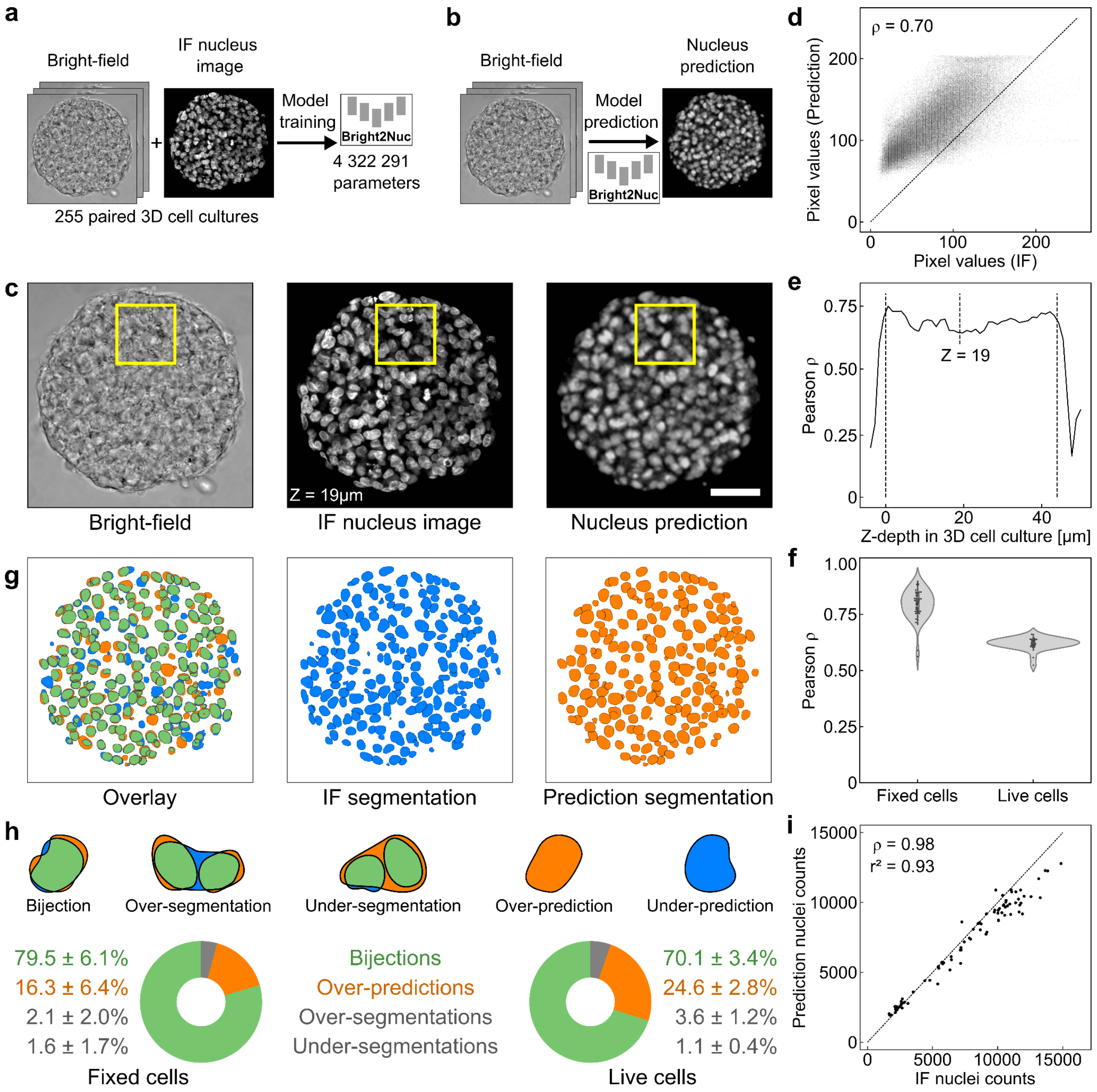


Figure 2 | Neural network accurately predicts nuclear staining in 3D cell cultures from bright-field images.

a, We trained Bright2Nuc, a U-Net-based neural network, on paired bright-field and IF nucleus images, and **b,** used it to predict nucleus images from bright-field images. **c,** Representative bright-field image of an on-chip 3D cell culture (left), corresponding ground-truth DAPI IF nucleus image (middle), and the predicted nucleus image (right). Yellow rectangles highlight the position of a specific region for comparison. Scale bar: 100 μm. **d,** Correlation between the fluorescence pixel values of the ground-truth nucleus image given in (**c**) and the pixel values of the corresponding predicted nucleus image (Pearson correlation ρ = 0.70). Each data point represents a single pixel value and the line represents the y = x identity. **e,** The model's prediction performance is independent of the imaging depth. The Pearson correlation between the predicted and ground-truth pixels was stable along the z-axis of the cell culture, as shown in (c). **f,** Average Pearson correlation between the ground-truth and predicted pixel values from whole 3D cell cultures imaged in either fixed (N = 54) or live (N = 31) states. **g,** 3D segmentation of the IF nucleus image in (c) (middle, blue) and the predicted nucleus image (right, orange). The overlays of both segmentations show a high overlap (left, green; 80.6% overlapping area with the ground-truth segmentation). **h**, Top: Categories for comparison of predicted segmentation ground-truth IF segmentation. Bottom: Distribution of categories for predicted segmentations from fixed (N = 54) and live (N = 31) 3D cell cultures. **i**, High accuracy for the prediction of nuclei counts in the 85 fixed and live 3D cell cultures in the test set (Pearson correlation ρ = 0.98; coefficient of determination r^2 = 0.93). The diagonal line represents the y = x identity.

The pixel values of the predicted nucleus images were highly correlated with the fluorescent nucleus signals of the ground-truth images (**Fig. 2d**, Pearson correlation of ρ = 0.70 for the pixel values of the image shown in Fig. 2c), independent of the tissue imaging depth (**Fig. 2e**). In the test set, predicted nucleus images from fixed 3D cell cultures showed an average pixel correlation of ρ = 0.79 ± 0.06 (mean ± SD, N = 54) that dropped to ρ = 0.62 ± 0.03 (N = 31) for live 3D cell cultures (**Fig. 2f**).

Accurate segmentation of nuclei from bright-field imaging

The predicted nucleus images were segmented in 3D using a re-trained StarDist [13,14] model (**Fig. 2g**). To benchmark the results, segmented nuclei were classified into five categories: (i) bijections, that is, predicted nuclei overlapping with a single nucleus in the ground-truth image; (ii) over-segmentations, that is, two or more predicted nuclei overlapped with a single nucleus in the ground-truth image; (iii) under-segmentations, that is, single predicted nuclei overlapped with two or more nuclei in the ground-truth image; (iv) over-predictions, where predicted nuclei overlapped with no nuclei in the ground-truth image; and (v) under-predictions, where nuclei in the ground-truth existed with no overlap in the prediction image (**Fig. 2h**). In fixed cultures, the Bright2Nuc and StarDist approaches resulted in 79.5 ± 6.1% of the predicted nuclei as bijections (mean ± SD, n = 47224). An additional 2.1 ± 2.0% and 1.6 ± 1.7% of the predicted nuclei were correct, but over- or under-segmentated, respectively. However, the approach yielded 16.3 ± 6.4% of over-predictions, whereas conversely, 16.1 ± 6.6% of all ground-truth segmentations were under-predicted (**Fig. 2h**). Furthermore, prediction was enabled in live cultures at the cost of a slightly decreased accuracy of 70.1 ± 3.4% of bijections (**Fig. 2h**). The number of nuclei in each 3D cell culture was predicted with a high correlation to the groundtruth (**Fig. 2i**, ρ = 0.98; r^2 = 0.93).

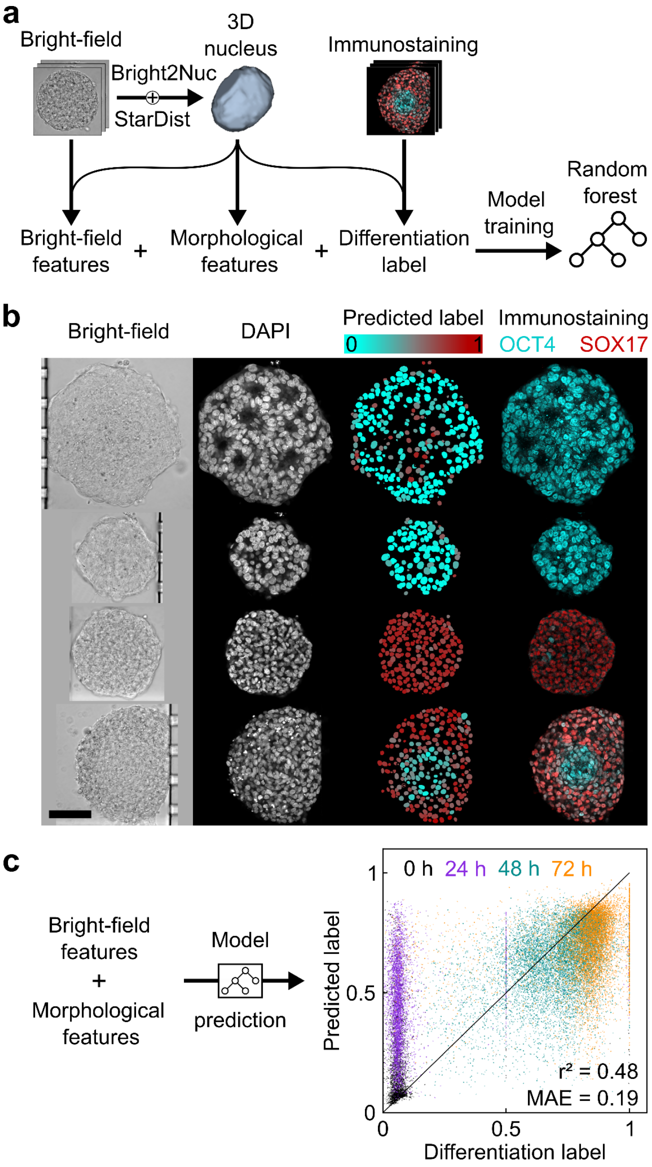


Figure 3 | Bright-field and predicted morphological features predict single-cell differentiation state. **a**, We extracted 56 features from bright-field and 63 morphological features from segmented Bright2Nuc 3D images in combination with the differentiation label obtained from the IF expression to train the random forest model. **b**, Visually, the predicted label (3rd column, color coded from cyan to red) matches the true IF expression (last column, merged IF images showing OCT4 in cyan and SOX17 in red). Scale bar: 100 µm. The random forest model predicted a differentiation label for each cell in the on-chip 3D cell cultures. **c**, Comparison of the ground truth to the predicted differentiation label is shown on the right. Each data point represents a single nucleus (*n* = 26213) extracted from 49 3D cell cultures. (r^2 = 0.48, mean absolute error = 0.19). The mean absolute error observed on nuclei in 3D cell cultures fixed 24 h after the start of DE differentiation is much higher (MAE_{24h} = 0.38 ± 0.18, *n* = 4376) than the one for other time points (MAE_{0h} = 0.15 ± 0.18, *n* = 3028; MAE_{48h} = 0.13 ± 0.11, *n* = 11364; MAE_{72h} = 0.17 ± 0.14, *n* = 7445). This discrepancy can be explained by the inaccurate value of the differentiation label obtained from the IF images. At 24 h, cells displayed high OCT4 and low FOXA2/SOX17 expression, thus resulting in low differentiation label values, even though the cells were already committed to the DE differentiation path. The random forest interestingly predicts a label value of DL_{24h}-pred = 0.46 ± 0.18, which spans the range between the values for the 0 h and 48 h time points (DL_{0h}-pred = 0.21 ± 0.18; DL_{48h}-pred = 0.64 ± 0.15).

Label-free prediction of differentiation state

Next, we investigated whether morphological and other features derived from bright-field images were predictive of hiPSC differentiation. In previous studies, the morphology of stem cell nuclei was shown to be indicative of their differentiation state [16–19]. Therefore, we first mapped the expression of the three TFs to a single normalized value: the differentiation label (DL). The DL was defined as the ratio between the average of the normalized expressions of e_{FOXA2} and e_{SOX17} of the differentiation markers FOXA2 and SOX17, and the sum of the normalized expression of e_{OCT4} of the pluripotency marker OCT4 and the differentiation markers:

$$DL \ = \frac{1}{1+\frac{2\,e_{OCT4}}{e_{FOXA2}+e_{SOX17}}}.$$

DL progressed from values close to zero for pluripotent cells at t = 0 h with high OCT4 expression to a value of ~1 for DE cells after 72 h with high FOXA2 and SOX17 expression (**Supplementary Fig. 4**). To predict the DL for each nucleus, we designed an explainable feature-based approach. For each of the 26213 segmented nuclei along the DE differentiation trajectory, 120 features were extracted: 64 morphological features from the 3D segmentations after Bright2Nuc and StarDist applications and 56 bright-field texture features were extracted from a fixed-sized bounding box centered on the nucleus on the paired bright-field 3D images. A random forest algorithm with 1000 estimators and a depth of 10 was trained on the features for predicting DL (**Methods**). The ground truth DL for model training was calculated for each nucleus from the TF expression in the bounding box of the associated IF image (**Fig. 3a**).

The resulting random forest model predicted single-cell DL in the bright-field images of 3D cell cultures fixed along the DE differentiation trajectory. In a round-robin fashion, five random forest models were trained on different training and test sets, ensuring that each cell culture was contained in the test set exactly once, while the training and test sets were split on the level of whole cell cultures. Visually, the predicted DL correlated with true IF staining, even in the case of non-homogeneously expressing cell cultures (**Fig. 3b**). For all 26213 nuclei, the predicted DL correlated with the true DL with a coefficient of determination of $r^2 = 0.48$ and a mean absolute error of 0.19 ± 0.17 (mean ± SD, **Fig. 3c**). We observed strong deviations for nuclei fixed 24 h after DE induction, where the mean absolute error was higher than that at other time points. Interestingly, the predicted DL values at 24 h spanned the entire range, indicating that the random forest predicted nuclei as being in a developmental transition state rather than resembling pure DE or pluripotent cell states. The discrepancy between the predicted and true DL can be explained by an inaccurate ground truth at 24 h, where cells exhibited a high OCT4 and a low FOXA2/SOX17 expression (**Fig.1d-e**), resulting in a low true DL (0.09 ± 0.14, mean ± SD, *n* = 4376) indicative of pluripotent cells, which does not reflect the state of those cells after 24 h of DE induction. In fact, a single-cell transcriptomic time trajectory of DE differentiation under comparable conditions [20] showed that after 24 h, the transcriptome of hiPSCs clustered differently from pluripotent stem cells (**Supplementary Fig. 5a**), while still expressing Oct4 at the mRNA level (**Supplementary Fig. 5c**). Consequently, the simplified view of cell type differentiation based on single-cell state fluorescence markers argues for descriptive states, whereas the neural network learning approach resolves a continuous cell type transition, as is seen with time-resolved single-cell transcriptomics. Our approach highlights the rich and hidden information sources of bright-field images and their potential to resolve the transcriptional states of human stem cells.

Label-free tracking of single cells

Bright2Nuc can bridge the domain gap between fixed and living tissues by tracking nuclei in live 3D cell cultures. To this end, we first generated ground-truth tracking data with 3D cell cultures formed from a mixture of wild-type hiPSCs and the SOX2-T2A-tdTomato reporter line in a one-to-two ratio to generate heterogeneous mixtures of labeled and non-labeled nuclei. Bright-field and IF images were acquired every 7.5 min for 17 h, resulting in 136 frames. The trained Bright2Nuc was then applied to every frame of the bright-field image sequence to predict nuclei. Predicted and IF-stained nuclei were tracked using TrackMate [21]. Expectedly, we detected roughly two thirds of the number of nuclei (**Fig. 4a,** blue curve) in the IF images as compared to nuclei from the predicted images (**Fig. 4a,** black curve). With increasing experimental time, the number of detected tracks increased due to cell proliferation in both curves, with the ratio between ground-truth and predicted tracks remaining constant at $n_{\mathrm{GT}}/n_{\mathrm{Pred}}$=0.58 ± 0.03 (**Fig. 4a**, mean ± SD, *n* = 136 frames). Assigning nuclei to tracks resulted in a more stable nuclei count than counting segmented nuclei alone (**Fig. 4a**; orange curve). Tracks in the ground truth are, on average, longer than tracks found in the predicted sequence (**Supplementary Fig. 6b**), which can be due to under-predictions of Bright2Nuc (**Fig. 2**), causing nuclei to be lost in a track.

For further tracking evaluation, we calculated the percentage of ground-truth tracks with a matched predicted track as a function of the predicted track length (**Fig. 4b**). We counted a match between two tracks if the distance between the center of mass of nuclei in the ground-truth and predicted images was less than 11 µm, which is roughly equal to the average nuclei diameter. The percentage of matched tracks decreased steadily from an initial 85% for the minimum requirement of two frames to approximately 50% for the requirement of 25 consecutive frames, corresponding to 3 h of imaging (**Fig. 4b**).

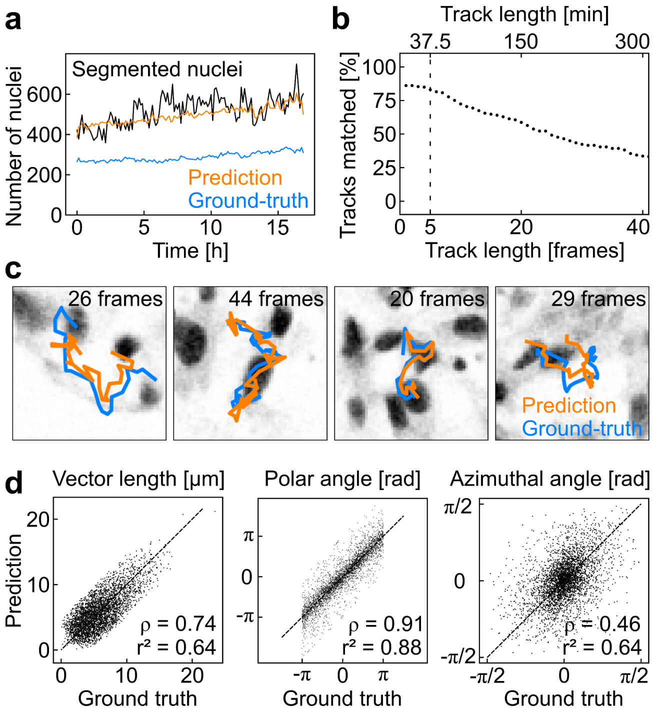


Figure 4 | Label free single-cell tracking in live 3D cell cultures.
a, Tracked nuclei in the label-free predicted images (orange) and fluorescence reporter images (blue) in one representative 3D cell culture. As per the ratio of fluorescent cells to the wild type, the ground-truth contained fewer tracks than the prediction. Incorporating temporal information by counting the number of tracks results in a more stable nucleus count than that based on the segmentations of each individual segmented label-free predicted image (black). **b**, If we require the tracks to match for a higher number of frames, the percentage of tracks in the fluorescence reporter images with a matched track in the predicted images decreases. The longer we require the tracks to be, the fewer we can identify them as matching. **c**, Representative 3D single-cell tracks in the predicted nucleus images (orange) and ground-truth SOX2-T2A-tdTomato fluorescence nucleus reporter images (blue). The background gray-value images show the ground-truth nuclear signal at the tracking end. The frame interval is 7.5 min. **d**, In spherical coordinates averaged over five frames, we found a high correlation in displacement (vector length, ρ = 0.74, r² = 0.64) and directionality (polar angle, ρ = 0.91, r² = 0.88, and azimuthal angle, ρ = 0.46, r² = 0.64) and between prediction and ground truth.

Finally, we compared tracks from the ground truth with their matched predicted tracks, taking an arbitrary threshold of five consecutive frames for which 81% of the ground-truth tracks had a predicted track match (**Fig. 4b**). Exemplary velocity vectors were derived by averaging the cell movement over five consecutive frames matched visually (**Fig. 4c**), and their spherical coordinates were correlated between the ground truth and predicted tracks with Pearson correlations for the vector lengths, polar angles, and azimuthal angles of ρ = 0.74, ρ = 0.91, and ρ = 0.46, respectively (**Fig. 4d)**. The distributions of both the ground truth and predicted velocity vector coordinates were similar (**Supplementary Fig. 6d-f**). Movement in the xy-plane was isotropic with a homogeneous distribution of the polar angle (**Supplementary Fig. 6e**), whereas the values of the azimuthal angle remained close to 0 (**Supplementary Fig. 6f**), indicating little movement in the z-direction. The limited z-directed movement was probably caused by the constrained geometry of the on-chip cultures. However, this confinement did not affect nuclei velocities, as evidenced by the lack of dependence of the velocity vector length on the z-position of the nuclei (**Supplementary Fig. 6c**). Overall, considering a track length of five frames, Bright2Nuc coupled with the TrackMate algorithm allowed us to track 81% of the nuclei in 3D cell culture (**Fig. 4c**).

Label-free detection of cell velocities during differentiation

During DE differentiation, human pluripotent stem cells undergo synchronous epithelial-mesenchymal transition (EMT) [22,23,24]. It is known that DE cells show significantly higher migration in 2D experiments as compared to pluripotent cells in a scratch assay [23], which argues that DE cells exhibit a higher cell mobility than stem cells. Therefore, in the last step, we investigated whether Bright2Nuc in combination with single-nuclei tracking can detect motility changes in label-free 3D cell cultures during DE differentiation. Along this line, we detected changes in the expression of cell adhesion molecules indicative of EMT during the DE differentiation on-chip (**Fig. 5a**). IF images of fixed 3D cell cultures along the DE differentiation trajectory showed that E-CAD/N-CAD expression changes co-occurred with upregulation of the DE marker SOX17. The single-cell transcriptomic dataset [20] also confirmed EMT TF changes, such as SNAIL1/2, whose expression peaked between the 24 h and 48 h time points (**Supplementary Fig. 5c**). To assess changes in cell motility during DE differentiation, we captured bright-field movies of six 3D cell cultures every 24 h over a period of 75 min with a 5 min/frame acquisition period (**Fig. 5**). Bright-field images were acquired using confocal microscopy with the same resolution as before. The Bright2Nuc + TrackMate approach was then used to extract the nuclei velocities, defined as the effective displacement of the nuclei over five frames (**Fig. 5a**). Strikingly, we observed an increase in effective cell displacement up to 48 h after induction of the DE differentiation. The highest effective cell displacement (D_{48h} = 0.98 ± 0.30 µm/min, mean ± SD) coincided with the time point of the E-CAD/N-CAD expression change and the appearance of the DE-specific marker SOX17. The population median of the four effective cell displacement distributions (D_{0h} = 0.78 ± 0.22 µm/min, D_{24h} = 0.70 ± 0.26 µm/min, D_{48h} = 0.98 ± 0.30 µm/min, D_{72h} = 0.82 ± 0.24 µm/min, mean ± SD) differs significantly ($p = 3.9 \times 10^{-242}$, Kruskal–Wallis H-test). Post-hoc analysis with a Bonferroni correction revealed a significant increase at the 48 h time point compared to the other time points ($p_{0h\text{-}48h} = 7.3 \times 10^{-165}$, $p_{24h\text{-}48h} = 1.4 \times 10^{-164}$, $p_{48h\text{-}72h}$

= 2.3 × 10^{-109}, Mann–Whitney-U-Test). Thus, the cells displayed higher motility during DE differentiation than during the pluripotent state. Notably, cell movements within the 3D cell cultures were non-directed throughout differentiation, as evidenced by the homogeneously low values of directionality calculated from the same velocity vectors (**Supplementary Fig. 7**). Interestingly, the measured increase in cell migration rates during DE differentiation was lower than that previously reported in 2D cultures [23], where DE cells showed a five times higher migration rate than that of pluripotent cells. This could reflect the more complex ECM within the 3D cell culture compared with a simple 2D cell surface culture.

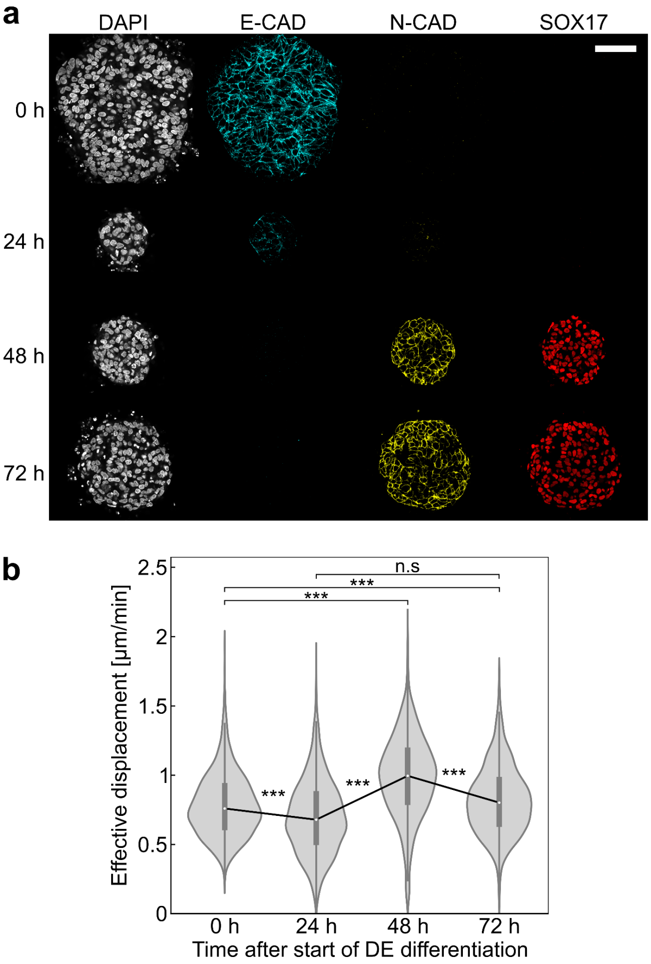


Figure 5 | Single-cell live tracking in 3D cell cultures during DE differentiation captures increased cell mobility.

a, Distribution of the effective displacement over five frames, calculated using Bright2Nuc and TrackMate on bright-field movies with a 24 h interval along the definitive endoderm differentiation trajectory. Each time point represents the distribution of the velocity vectors of nuclei in six 3D cell cultures (number of nuclei: n_{0h} = 2862, n_{24h} = 1308, n_{48h} = 2144, and n_{72h} = 2820). Movies were acquired with a 5 min per frame interval over 75 min. White circles indicate the mean, boxes mark the first and third quartiles, and bars indicate standard deviation. **b**, A significant increase in effective displacement correlates with the change in cadherin expression, as shown in the immunofluorescence confocal images of 3D cell cultures fixed along the DE differentiation trajectory, showing the evolution of epithelial-cadherin expression (E-CAD) and neural-cadherin expression (N-CAD). The evolution of the differentiation marker SOX17 was shown to mark progress towards the DE stage. Scale bar: 100 µm.

Discussion

Microfluidic cell culture technologies for controlling the massive parallelization of 3D cell cultures are rapidly advancing, while high-throughput analytical methods to exploit miniaturized biological samples are lacking. In this study, we provide a label-free and live imaging approach to cope with the faster image acquisition of on-chip-cultivated 3D stem cell cultures. The developed neural network learning algorithm, Bright2Nuc, can be coupled with pre-existing image analysis tools such as StarDist and TrackMate for segmentation and tracking, respectively. With StarDist, Bright2Nuc enabled us to infer the cell number and nuclear locations from accessible confocal bright-field images with an accuracy of over 80% for 3D cell cultures with more than 10000 cells. Using the simplest imaging methods for high-content imaging, we offer a general imaging approach for screening 3D tissues. Deep learning approaches to detect cell nuclei have been previously employed for 2D adherent cell cultures [8,9,11,12] or histological tissue slices[10]. Existing neural networks have not yet attempted to infer information from cell images acquired with lower numerical aperture objectives and xy-resolution[9–11] in 3D. Beyond locating only the nucleus volume and positions, morphological features from inferred nuclei and corresponding bright-field images allowed the determination of transcriptional cell states along the pluripotent to DE differentiation trajectory. Notably, the resulting cell state predictions were more continuous than the cell state descriptions generated from the fluorescence signals of descriptive fluorescence markers, which is the standard research approach in developmental biology. Our results highlight the currently untapped potential information contained in the bright-field images.

In addition to the static view, we demonstrated that Bright2Nuc can resolve real-time information from live nuclei in 3D cell cultures by using a neural network together with the TrackMate algorithm. For example, we captured label-free cell dynamics, indicating epithelial-mesenchymal transition on a timescale of hours. The homogeneous transition from pluripotent to DE cell stage within the on-chip cultivated 3D cell culture was reflected in the nuclei velocity changes during differentiation. Nevertheless, the provided image analysis method will allow the assessment of motion directionality, cell migration, and neighboring cell-cell contacts upon formation of architecturally complex structures into endodermal tissue, such as pancreatic or liver cell types. To further decrease the image acquisition time and increase the time resolution for subsequent 3D cell cultures, we expect that Bright2Nuc can be adopted for widefield bright-field images. In summary, we believe that the general framework of Bright2Nuc opens a wide field of applications for the high-content screening of 3D cell cultures on chips.

Code availability

Bright2Nuc is available as a pip-installable package, together with commented analysis scripts at https://github.com/marrlab/Bright2Nuc.

Data availability

The dataset is made available via Zenodo

Author contributions

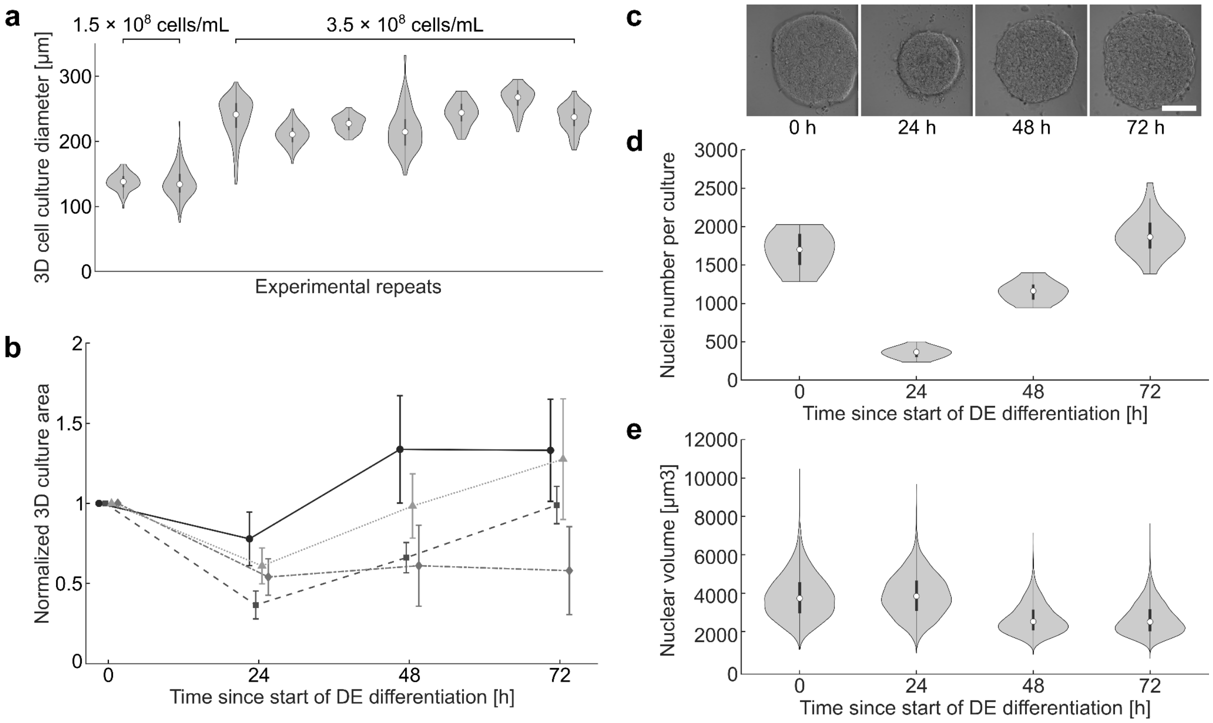
SA cultured, stained, and imaged the cell lines. DW implemented the code, trained the models, and performed the computational experiments. SA and DW evaluated the experiments, designed the figures, and wrote the manuscript with CM and MM. DW and SSB developed a Bright2Nuc Python package. CM and MM supervised this study.

Acknowledgements

We thank Fabian Theis, Henrik Semb, Sophia Wagner, Melanie Schulz, Benedikt Mairhörmann, and Valerio Lupperger for their discussions and ideas.

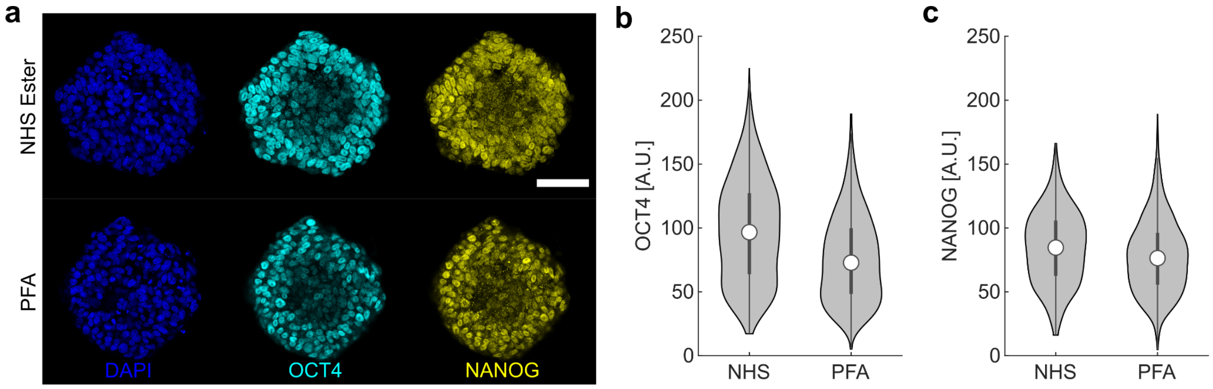
Funding

CM and MM received funding from the European Research Council (ERC) under the European Union’s Horizon 2020 research and innovation program with Grant agreement No. 866411 and 772646, respectively. This work was funded by the Helmholtz Pioneer Campus. SSB has received funding by F. Hoffmann-la Roche LTD (no grant number applicable) and supported by the Helmholtz Association under the joint research school ‘Munich School for Data Science - MUDS’.



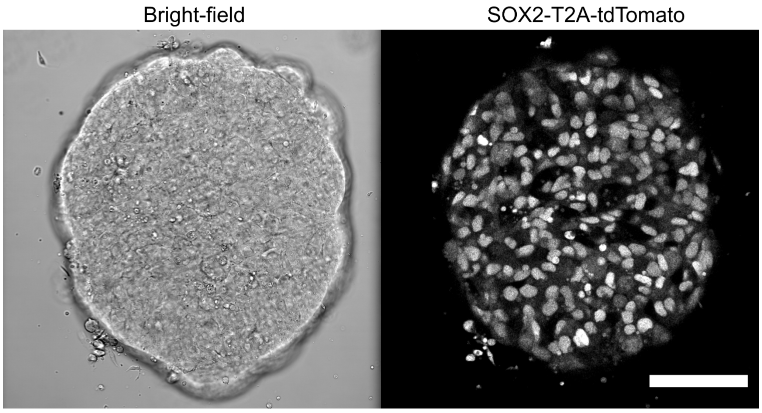
Supplementary Figure 1 | On-chip 3D cell culture and hiPSC differentiation.

a, Distribution of 3D cell culture diameters measured 24 h after seeding cells on the microfluidic platform across nine experimental replicates. The first two repeats were seeded with a single-cell solution with a concentration of 1.5 × 10^8 cells/mL; and the last seven repeats were seeded at a concentration of 3.5 × 10^8 cells/mL. White circles indicate the mean, boxes mark the first and third quartiles, and bars indicate standard deviation. **b**, Evolution of the area occupied by 3D cell cultures measured in the xy-plane every 24 h along the DE differentiation trajectory. Cell culture areas were normalized individually to the area measured before the start of differentiation. Data points represent the mean and error bars show the standard deviation. Each of the four curves represents a distinct biological replicate with N = 128 cell cultures. **c**, Bright-field images of a typical 3D cell culture taken every 24 h along the DE differentiation trajectory. **d**, Evolution of the number of nuclei contained in 3D cell cultures fixed every 24 h along the DE differentiation trajectory for one experiment (curve with squares in **b**) with the corresponding evolution of the nuclear volume (**e**). White circles indicate the mean; boxes mark the first and third quartiles; bars indicate the standard deviation (N_{0h} = 16 cultures, N_{24h} = N_{48h} = N_{72h} = 36; n_{0h} = 26973 nuclei, n_{24h} = 12802, n_{48h} = 41550, and n_{72h} = 67780).



Supplementary Figure 2 | Comparison of fixation methods to enable time trajectories on the mLSI chip.

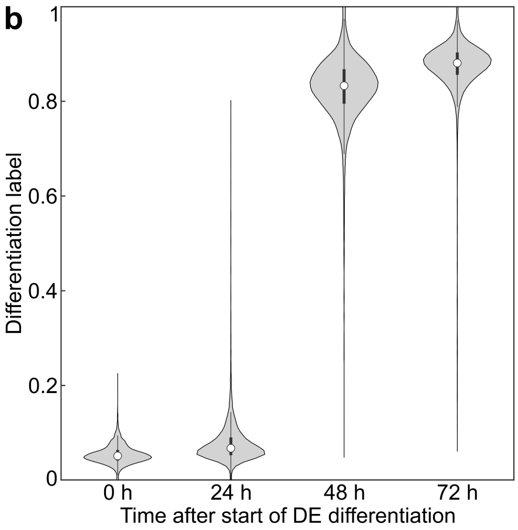
a, Immunostaining images of iPSC 3D cell cultures stained for pluripotency markers OCT4 and NANOG. The top cell culture was fixed using 40 mM Bis-PEG4-NHS ester, while the bottom was fixed using 4% PFA. Scale bar: 100 µm. **b-c**, Mean fluorescence [A.U.] for OCT4 and NANOG in individual nuclei measured from confocal images after segmenting the DAPI signal. The distribution of nuclei fluorescence signals is shown for the 3D cell cultures fixed either with NHS (*n* = 1706 nuclei in eight cell cultures) or with PFA (*n* = 2852 nuclei in 10 cell cultures). White circles indicate the mean, boxes mark the first and third quartiles, and bars indicate standard deviation.



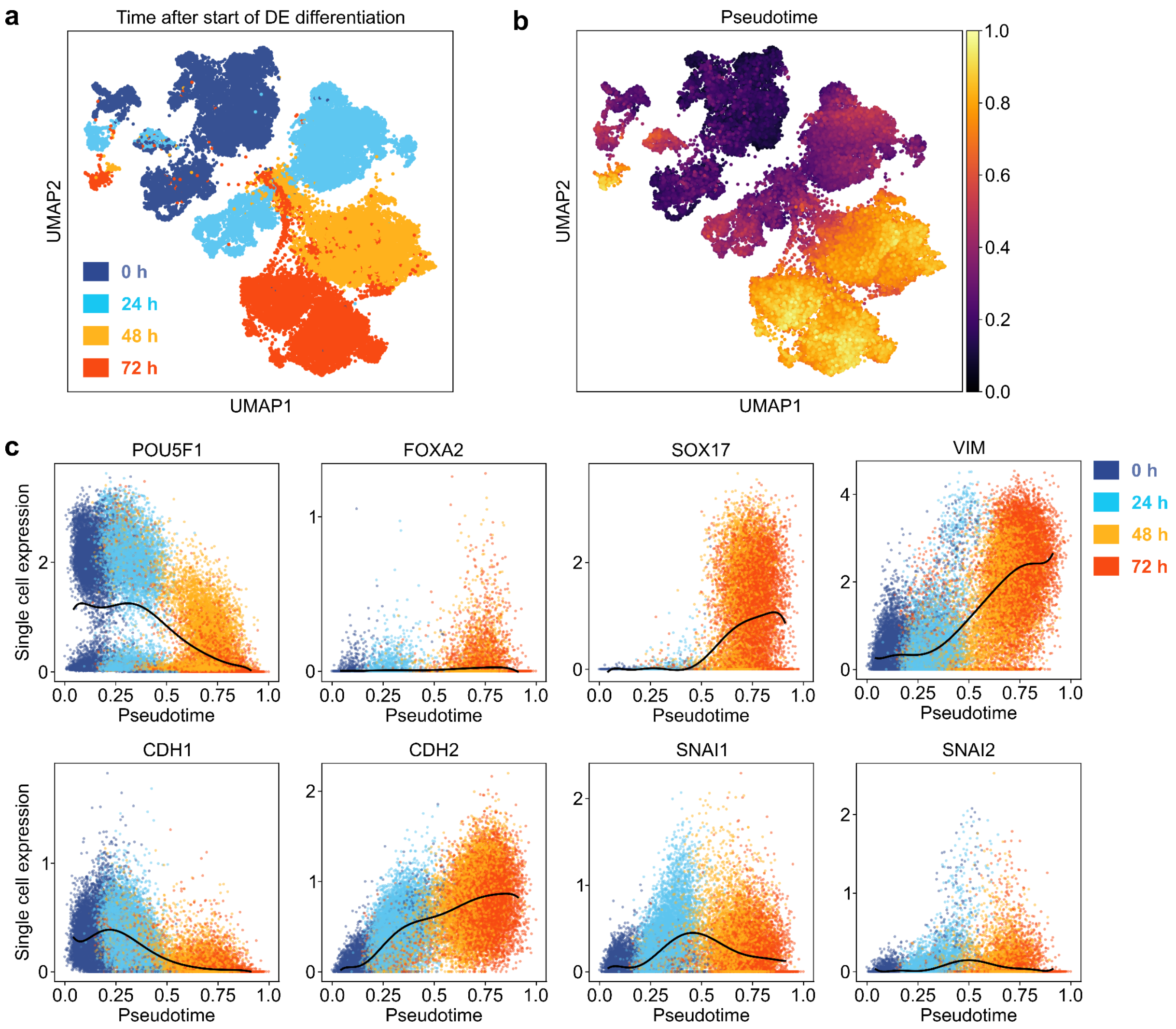
Supplementary Figure 3 | Confocal image of a pluripotent SOX2-T2A-tdTomato-reporter live 3D cell culture. Scale bar: 100 µm.

a

$$\text{Differentiation label} = \frac{\dfrac{e_{SOX17} + e_{FOXA2}}{2}}{e_{OCT4} + \dfrac{e_{SOX17} + e_{FOXA2}}{2}}$$



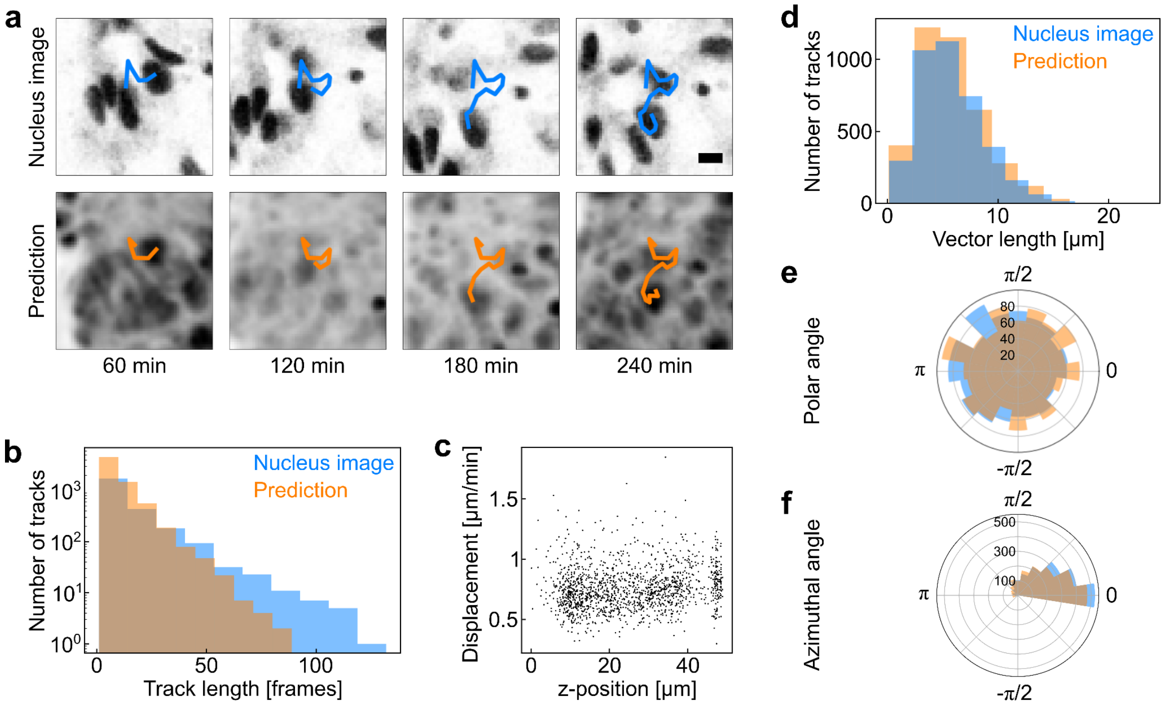
Supplementary Figure 4 | The differentiation state of cells expressed as differentiation label.
a, The differentiation label is defined as the ratio of the average normalized expression of the differentiation markers e_{FOXA2} and e_{SOX17} to the sum of pluripotency (e_{OCT4}) and differentiation expression. **b,** Evolution of the differentiation label along the definitive endoderm differentiation trajectory for the experiment presented in Fig. 1 (nuclei considered: n_{0h} = 5501, n_{24h} = 8700, n_{48h} = 22587, n_{72h} = 14936; 3D cell cultures: N_{0h} = 12, N_{24h} = 25, N_{48h} = 24, and N_{72h} = 30). White circles indicate the mean, boxes mark the first and third quartiles, and bars indicate standard deviation.



Supplementary Figure 5 | Single-cell transcriptomics of hiPSCs during definitive endoderm differentiation.

Data adapted from datasets published by Cuomo et al. [20]

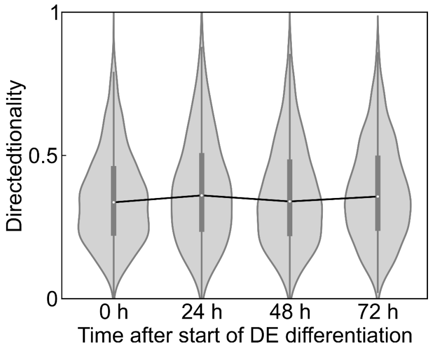
a, UMAP clustering of single-cell RNA expression of human induced pluripotent stem cells sampled every 24 h during a 72 h differentiation protocol towards definitive endoderm. The overlaid colors represent the sampling time after the start of the DE differentiation. **b,** UMAP clustering overlaid with pseudotime, a measure of the differentiation state of the cells towards DE, analogous to our differentiation label. **c,** Single cell expression of selected genes as a function of pseudotime; colors indicate the sampling time. POU5F1, CDH1, and CDH2 are known as OCT4, E-CAD, and N-CAD, respectively. VIM is a mesenchymal marker that typically appears after the EMT. SNAI1 and SNAI2 are markers closely associated with EMT; interestingly, their expression peaks exactly between 24 h and 48 h.



Supplementary Figure 6 | Single-cell live tracking in 3D cell cultures.

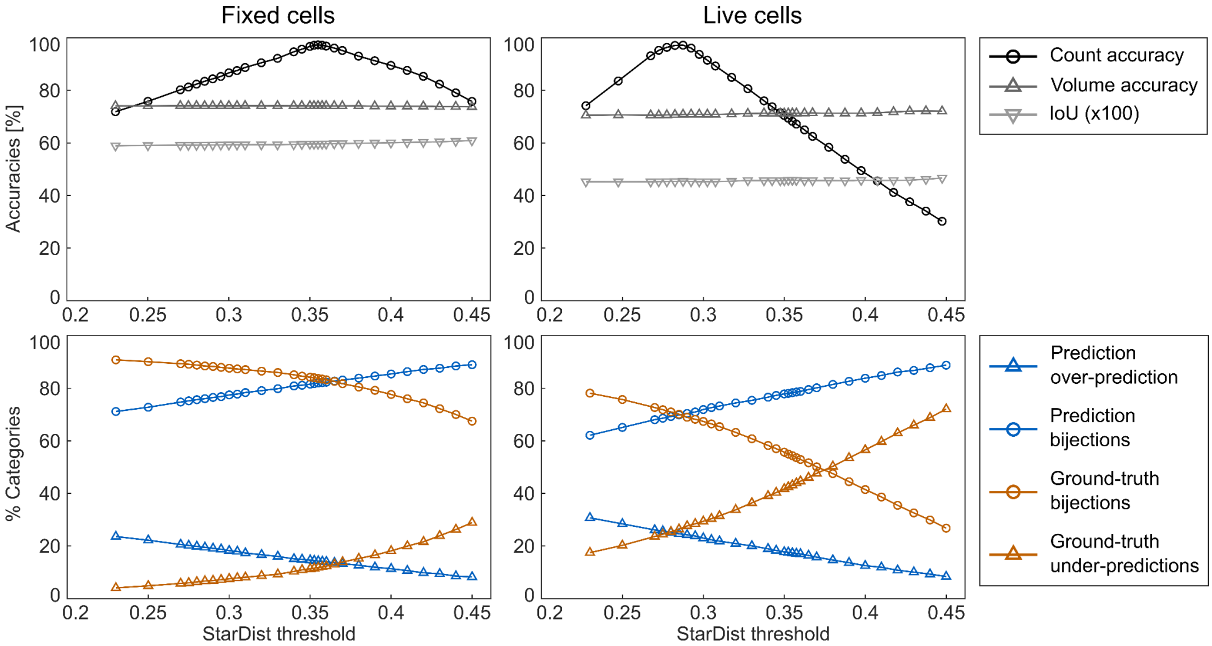
a, Example frames with a 60 min interval for the tracking of a nucleus using both the ground-truth nucleus image (top row, SOX2 expression of pluripotent iPSCs) and *in silico* prediction from Bright2Nuc (bottom row). The overlaid tracks can be seen to have similar shapes in the ground truth and prediction. Scale bar: 5 µm. **b,** Distribution of track lengths. More vectors with a shorter length were found within the Bright2Nuc image frames than for the nucleus image tracks, as the tracking has a longer temporal persistence in the ground truth. **c,** Displacement of individual nuclei as a function of their z-position in the 3D cell culture. There was no observed dependence on the displacement relative to the z-position. **d-f,** Distribution of velocity vector coordinates for vectors with a mean error smaller than 11 µm between the positions of the nuclei in the prediction and the nuclear signal. Both distributions were remarkably similar in length (d), polar angle (e), and azimuthal angle (f). The mean velocity vector was calculated for five consecutive frames for 5362 corresponding nuclei.

$$\text{Directionality} = \frac{\text{Effective distance}}{\text{Total distance}}$$



Supplementary Figure 7 | Directionality of cell movement in live tracking of 3D cell cultures during DE differentiation.

Directionality, defined over five frames as the effective displacement divided by the absolute value of the displacement, is a measure of the randomness of cell movement. The directionality is calculated using Bright2Nuc and TrackMate on bright-field movies from Fig. 5 with a 24 h interval along the definitive endoderm differentiation trajectory. Each time point represents the distribution of the velocity vectors of nuclei in six 3D cell cultures (n_{0h} = 2862, n_{24h} = 1308, n_{48h} = 2144, and n_{72h} = 2820). Movies were acquired for 5 min. frame intervals over 75 min. White circles indicate the mean, boxes mark the first and third quartiles, and bars indicate standard deviation.



Supplementary Figure 8 | Threshold optimization for StarDist segmentation.

The probability threshold, inherent to the StarDist model, for the segmentation of the *in silico* prediction nucleus image was optimized by maximizing the nuclei count accuracy calculated by comparing the number of objects segmented in the prediction with the number of objects in the ground-truth segmentation. Objects determined by the StarDist algorithm are accepted as valid segmentations if they have a probability that is higher than this threshold. Consequently, a higher threshold indicates fewer objects, but these objects are generally more trustworthy. Indeed, as the threshold increases, the percentage of over-predicted objects diminishes, and bijections increase as a percentage of *in silico* predictions. However, as fewer objects are predicted, the percentage of under-predictions increases as the bijections decrease as a percentage of ground truth objects. The value of the threshold has little to no consequence on the mean volume accuracy of the predicted bijections or their intersection over union ratio (IoU). It should be noted that, as a result, different threshold values were used when segmenting *in silico* prediction nucleus images of fixed cell cultures (threshold = 0.355) and live cell cultures (threshold = 0.290).

Supplementary Video 1 | Seeding process of the microfluidic platform.
Seeding of single-cell suspensions in the chambers of the microfluidic large-scale integration chip platform. Cells were perfused into each of the individually addressable cell culture chambers while the pneumatic membrane valves were in the open state. Cells in the four central cell compartments were shielded from direct flow upon the actuation of the pneumatic membrane valves. Subsequently, side channels were rinsed and the cell culture process, including a feeding cycle of every 4 h for 30 s, was started. Video dimension: 5570 × 3324 μm². The video speed was real-time.

Supplementary Video 2 | Live tracking of nuclei in pluripotent SOX2-T2A-tdTomato-reporter 3D cell culture.
Video of a pluripotent SOX2-T2A-tdTomato-reporter 3D cell culture imaged with a confocal microscope with a 20X/0.8NA objective, as shown in Fig.4. Cell culture was imaged every 7.5 min for 17 h. (Left) Bright-field image of the 3D culture at a height of 5 μm. (Right) Average RFP fluorescence signal in the axial direction of the first 10 μm of the 3D cell culture. The nuclei in this video were tracked in 2D using StarDist integrated with TrackMate in ImageJ. The tracks are color-coded, and the lines indicate the past trajectories of the nuclei. Video height: 350 μm. Video speed:8 frames/s i.e. 3600×.

Methods

Microfluidic chip fabrication

PDMS chips were produced using traditional soft lithography techniques for two-layered devices [25]. Briefly, two molds were fabricated by photolithography. The mold for the control channel network was made to have features 25µm in height using SU-8 3025 (MicroChem). The mold for the flow flayer was fabricated in three steps. First, the flow channel was made using the AZ40XT photoresist (MicroChemicals) at a height of 40µm and re-flowed to obtain rounded half-channels. Second, the perfusion channels of the cell culture chambers were produced at a height of 20µm using SU-8 3025 (MicroChem). Finally, the cell chambers were made using SU-8 3050 at a height of 50µm. All masks for photolithography were designed using the AutoCAD software (AutoDesk, 2019) and photoresists were patterned using a laser Micro Pattern Generator (µPG101, Heidelberg Instruments). All molds were coated with CYTOP^{TM} (CTL-809M; AGC Chemicals) to prevent the adhesion of the PDMS. Chips were cast from the molds using Sylgard 184 PDMS (Dow Corning). The chips were assembled in a push-down configuration with the flow layer on the bottom. Flow and control layers were bonded using the off-ratio PDMS bonding procedure (5:1 pre-polymer to cross-linking reagents ratio for the control layer and 20:1 ratio for the flow layer). Finally, the flow layer was bonded to a glass slide carrier (Brain Laboratories) through oxygen plasma activation (20 W at 0.9 mbar for 25 s). Each cell culture chamber comprised a central flow channel and two side channels bifurcating at the entry of the culture chamber. Pneumatic membrane valves divided the central flow channel into four 640x400µm² cell culture compartments with a volume of 0.013 µL. Gaps with a cross-section of 20x20µm² allowed the crossing of fluids between the side channels and the culture compartments.

Microfluidic chip operations

Pneumatic microfluidic valves (PMVs) within the chips were operated by applying a 1.5 mbar pressure on the control lines. The pressure could be switched on and off automatically on each individual control line using a homemade system enabling the individual control of 24 solenoid valves (LMV155RHY-5A-Q; SMC). Additionally, the system included a pressure regulator (ref pressure regulator) connected to 8 additional solenoid valves which applied a controllable pressure (0-1.5 bar) on light-proof gas-tight bottles containing the reagents to be introduced in the chip. All connections were made using Tygon tubings (ND 100-80; Proliquid, Germany). Chips were placed in a microscope stage top incubator (STX, Tokai Hit®) for maintaining a 37°C and 5% CO_2 humidified environment while allowing live imaging. Prior to cell seeding, the cell culture chambers were coated with a 10% Pluronic® F-127 (Cat#9003-11-6; Sigma-Aldrich) in phosphate-buffered saline (PBS) overnight to prevent cell adhesion to the glass substrate or PDMS walls. Adherent cells were harvested at 70-80% confluence with TrypLE Express (Cat# 12604013; Gibco) and resuspended in 40-60 µL at a 1.5-3.5 × 10^8 cells/mL in maintenance medium (mTeSR^{TM} Plus; StemCell Technologies) supplemented with 10µM ROCK inhibitor (Y-27632, Cat# sc-281642A; Santa Cruz Biotechnology). For the formation of 3D cell cultures, a homogeneous single-cell solution was introduced into each cell culture chamber with PMVs in the open state. Upon actuation of the PMVs in the cell culture chamber, cells in the four compartments were isolated from the main flow stream, and cells in the side channels could be rinsed. Once seeded, cells were left to self-aggregate for 4 h before the first media renewal; media was then renewed in the cell culture chambers every 2 h using a 100 mbar forward fluidic pressure. The average diameter of on-chip 3D cell cultures depended on the single-cell solution density, i.e., 138 ± 13 µm for a cell density of 1.5 × 10^8 cells/mL and 243 ± 19µm for a cell density of 3.5 × 10^8 cells/mL. The 3D cell culture formation process after seeding was robust, as demonstrated by a low average coefficient of variation CV = 0.10 ± 0.05 and a chip-to-chip variation as low as CV = 0.08 (**Supplementary Fig. 1**).

Human induced pluripotent stem cells (hiPSC) culture

When not specified otherwise, experiments were conducted using a hiPSC cell line [26] which was kindly provided by Alexander Kleger from Internal Medicine I, University Hospital, Ulm, Germany. In some experiments for the acquisition of live fluorescence nuclear signals, a SOX2-T2A-tdTomato reporter hiPSC line was used, generated by Shahryari et al [27]. All cell lines were maintained in a pluripotent state as 2D adherent monolayers in conventional cell culture plates coated with Geltrex (Cat# A1413302; Life Technologies), fed daily using mTeSR^{TM} Plus maintenance medium (StemCell Technologies, Canada) and maintained in a humidified atmosphere at 37°C and 5% CO_2. After reaching 70–80% confluence, the cells were passaged with 0.5 mM EDTA (Cat# A4892; AppliChem) in PBS. To enhance cell viability after splitting, the maintenance medium was supplemented with 10 µM ROCK inhibitor (Y-27632, Cat# sc-281642A; Santa Cruz Biotechnology) for the following 24 h. Mycoplasma-free cell culture was regularly confirmed using a MycoAlert^{TM} Plus Mycoplasma Detection Kit (Cat# LT07-703; Lonza).

Definitive endoderm differentiation

Before DE induction, cells were fed for 24 h with maintenance medium and ROCK inhibitor and another 24 h with maintenance medium only. DE differentiation was induced 48 h after on-chip seeding, the start of differentiation has been called 0 h in this paper. We followed the differentiation protocol as published in previous literature [28], where the basal medium **(Supplementary Tables 1 and 2**) was supplemented with 100 ng/ml activin A (AA) (Cat# 120-14-300; Peprotech) and 3 µM CHIR-99021 (CHIR, GSK3β inhibitor, Cat# 24804-0004; Tebu-bio) on the first day of differentiation. Basal medium supplemented with 100 ng/ml AA and 0.3 µM CHIR was added on the second day, and basal medium with 100 ng/ml AA was added on the third day of differentiation.

Immunocytochemistry on-chip

Partial fixations of 3D cell cultures (fixation of a subset of cell cultures in a single microfluidic platform, while keeping other cell cultures alive) were performed using NHS Ester (Bis-PEG4-NHS ester, CAT# BP-21602; BroadPharma). At the end of all experiments, before immunocytochemistry, all cell cultures were fixed in 4% paraformaldehyde in deionized water for 1 h (even if previously partially fixed NHS ester). Cell membranes were permeabilized with 0.2% Triton X-100 and 100 mM glycine in deionized water for 6 h and blocked in a blocking solution containing 3% donkey serum, 10% fetal calf serum, 0.1% Tween-20, and 0.1% bovine serum albumin (BSA) in PBS. Primary antibodies were diluted in the blocking solution and incubated in the cell culture chambers for 24 h before being rinsed away with PBS. Secondary antibodies and DAPI were then diluted also in the blocking solution and incubated for 24 h. All cell culture chambers were thoroughly rinsed with PBS prior to imaging. All steps were conducted on-chip and at room temperature. A full list of the antibodies used is given in **Supplementary Table 3**.

Image acquisition

IF images and live bright-field images were acquired using a laser scanning confocal inverted microscope (Zeiss LSM 880 Airyscan) controlled by the ZEN Black version number software with a 20x/0.8-NA (numerical aperture) objective (Zeiss Plan-Apochromat 20x/0.8

M27), with up to five 8-bit or 16-bit data channels per image: transmitted light (bright-field using either the 633 nm or the 561 nm wavelength laser), DNA labeled with DAPI, and three types of antibodies distributed in the green (488 nm excitation), yellow (561 nm), and red (633 nm) channels (see **Supplementary Table 3** for antibody list). All images were acquired with a 0.25 × 0.25 µm² pixel size with the number of pixels being adjusted so that the scanning area could fit the imaged object in the frame (typical scanning areas were around 200 × 200 µm²). Confocal stacks were acquired to image the full 50 µm z-depth of the 3D cell cultures with a z-resolution of 1 µm. For the imaging of fixed tissue, the pixel dwell time was kept in the 1 - 2 µs range, depending on scanning area size, with a two-fold line averaging. For live tissue, pixel dwell times were reduced to the 0.6 - 0.9 µs range, to accelerate imaging so that a full confocal stack of a 3D cell culture could be acquired in less than four minutes.

Fluorescence signal correction

Prior to nuclear fluorescence signal quantification, two types of corrections were applied to the confocal image stacks. The first correction pertained to the weakening of fluorescence signals caused by the imaging depth in the 3D cell cultures. We calculated the relative average nuclear fluorescence signals (ratio between the average nuclear signal at a given depth and the average nuclear signal at the lowest point of the cell culture) as a function of the imaging depth. Under the assumption that nuclear signals should be constant on average relative to imaging depth, we calculated the imaging-depth decay rate through linear regression over all samples and for each wavelength. This wavelength-dependent correction was then applied to all image stacks. A second correction addressed antibody penetration inside the 3D cell culture during full-mount staining. Antibody penetration is a complex issue, which we resolved assuming that it factored in three main components: the antibodies, the x-y distance to the border of the cell culture, and the tissue type (differentiation state). We thus calculated the relative average nuclear fluorescence signal per antibody as a function of x-y distance to the border of the cell culture. The decay rates were calculated through linear regression per antibody for every sample within a time point to account for tissue type. All stacks were corrected under the assumption that nuclear signals should be constant on average relative to x-y penetration.

In-silico nuclear staining

Bright2Nuc is a deep learning framework with a modified 3D U-Net [29], based on the InstantDL [30] package, designed to predict nuclear markers from bright-field images (**Fig. 2a-b**). Bright2Nuc works on 3D data, predicting the nuclear marker from slice to slice. To incorporate 3D information from the bright-field image stack, we added nearest neighbor slices to the input of the network. The U-Net architecture was modified from the 3D U-Net used in InstantDL by keeping the 3D convolutional layers the same but using anisotropic max-pooling in the encoder, and anisotropic upsampling in the decoder, keeping the z-dimension unchanged. Bright2Nuc can be trained and evaluated using 3D stacks, through which it will automatically iterate during training, Bright2Nuc will crop or pad the input data to a size of 384 pixels in x-y dimension. During inference, it can handle arbitrary image sizes. If desired, it can rescale the images to keep the nuclei diameter constant, which simplifies transfer learning between datasets. An average nucleus diameter of 30 pixels worked best for our model. All considered images were bilinearly downscaled by a factor of 2 to a 0.5 µm/px xy-resolution before being processed for further analysis. We trained Bright2Nuc (i) on 162 fixed 3D cell culture images in 3D in bright-field and with a DAPI staining and (ii) on 93 cell cultures recorded in 3D in live tissue in bright-field and with a SOX2-T2A-tdTomato reporter signal. Together, we used approximately a quarter million nuclei/cells. For training, we optimized a mean squared error loss for 50 epochs with a batch size of 5. For data augmentation, we used brightness and contrast shifts of 30% of the pixel values, horizontal and vertical flips, and zooms with a maximum of 30% of the image size. The test set contained 85 cell cultures split in a stratified manner from the experimental data, summing up to around 80.000 cells/nuclei. Bright2Nuc is a ready-to-use python package, it can be downloaded via GitHub or installed using pip install and comes with our pre-trained model.

Nuclei segmentation

We trained two StarDist [14] 3D models, one to segment cell nuclei from images of a nuclear marker staining (stardist_nuc), and one to segment 3D cell cultures with an *in silico* staining (stardist_silico). We manually segmented five 3D cell cultures and trained stardist_nuc on the corresponding DAPI staining. To train stardist_silico, we created a training set of 15 3D cell cultures. We used the same five manually segmented cell cultures with the corresponding DAPI staining and the same five *in silico* stained cell cultures to the training set. Additionally, five *in silico* stained 3D cell cultures from the SOX2-T2A-tdTomato-reporter dataset were added, for which the segmentation ground truth was obtained by segmenting the SOX2-T2A-tdTomato signal using the first Stardist model trained on DAPI and manually verifying the results, summing up to 15 cell cultures. We trained the models as described in the StarDist documentation, but changed the anisotropy to two (x : 0.5, y : 0.5, z : 1) to match our imaging resolution. To adapt the StarDist model to the live tissue data, we found it sufficient to lower the StarDist threshold to 0.29 without retraining the stardist_silico model (**Supplementary Fig. 8**). With two resulting models (one for DAPI, one for in-silico) we segmented all DAPI stained 3D cell cultures and the *in silico* stained 3D cell cultures used in this paper.

Transcription factor expression prediction

To assess the differentiation status of single nuclei, we formulated the expression of the pluripotent marker OCT4 (e_{OCT4}), and the two differentiation markers FOXA2 (e_{FOXA2}) and SOX17 (e_{SOX17}) as a single ratio called differentiation label (DL) (**Supplementary Fig. 4**). Single-nuclei TFs expression were extracted from the corrected IF images (see Fluorescence signal correction in materials) by calculating the average fluorescence over each of the 3D DAPI StarDist segmentations. TFs expressions were normalized per dataset between 0 and 1 to the 1-th and 99-th percentiles, respectively. DL was then calculated for each nucleus as:

$$DL = \frac{1}{1 + \frac{2\, e_{OCT4}}{e_{FOXA2} + e_{SOX17}}}$$

From each segmented nucleus, 56 bright-field and 64 morphological features were extracted. The data was then randomly split into five folds, with an 80% / 20% ratio between train and test set, and a random forest regressor was trained with 1000 estimators on each fold. In total, five random forest models were trained using the scikit-learn [31] framework in a round-robin fashion, splitting 20% of the nuclei into test sets iteratively, so that each nucleus was contained in the test set once.

Live cell tracking

Single nuclei from the live *in silico* stained 3D cell cultures were tracked with Trackmate [21], using the integrated LoG detector with a sigma of 22 pixels (11 μm) and a quality threshold of 10. Data anisotropy was accounted for by setting the x- and y-values to 0.5 µm (leaving the z-value at 1 µm). The maximum linking distance was set to 22 pixels (11 µm) and the frame gap to two frames and 22 pixels. Tracking results were saved as a .csv file and evaluated in Python using pandas [32]. Predicted tracks outside the cell culture were filtered out using neighborhood-based filtering, removing all tracks that have less than 5 neighbors in a distance of 50 pixels per time point. Effective displacements were calculated as the effective distance over five frames divided by the time, with the effective distance being the distance between the centers of mass of the considered nucleus between the first and fifth frames. Directionality was also calculated over five frames as the ratio of the effective distance and the total distance, with the total distance being the sum of distances between the centers of mass and between all considered frames.

References

1. Avior, Y., Sagi, I. & Benvenisty, N. Pluripotent stem cells in disease modelling and drug discovery. *Nat. Rev. Mol. Cell Biol.* **17**, 170–182 (2016).
2. Takahashi, Y., Takebe, T. & Taniguchi, H. Engineering pancreatic tissues from stem cells towards therapy. *Regenerative Therapy* **3**, 15–23 (2016).
3. Zhao, X., Zhu, Y., Laslett, A. L. & Chan, H. F. Hepatic Differentiation of Stem Cells in 2D and 3D Biomaterial Systems. *Bioengineering (Basel)* **7**, (2020).
4. Liu, Y. & Chen, Y.-G. 2D- and 3D-Based Intestinal Stem Cell Cultures for Personalized Medicine. *Cells* **7**, (2018).
5. Luni, C., Gagliano, O. & Elvassore, N. Derivation and Differentiation of Human Pluripotent Stem Cells in Microfluidic Devices. *Annu. Rev. Biomed. Eng.* (2022) doi:10.1146/annurev-bioeng-092021-042744.
6. Lukonin, I., Zinner, M. & Liberali, P. Organoids in image-based phenotypic chemical screens. *Exp. Mol. Med.* **53**, 1495–1502 (2021).
7. Nie, J. & Hashino, E. Organoid technologies meet genome engineering. *EMBO Rep.* **18**, 367–376 (2017).
8. Lee, G., Oh, J.-W., Her, N.-G. & Jeong, W.-K. DeepHCS++: Bright-field to fluorescence microscopy image conversion using multi-task learning with adversarial losses for label-free high-content screening. *Med. Image Anal.* **70**, 101995 (2021).
9. Christiansen, E. M. *et al.* In Silico Labeling: Predicting Fluorescent Labels in Unlabeled Images. *Cell* **173**, 792–803.e19 (2018).
10. Rivenson, Y. *et al.* PhaseStain: the digital staining of label-free quantitative phase microscopy images using deep learning. *Light Sci Appl* **8**, 23 (2019).
11. Ounkomol, C., Seshamani, S., Maleckar, M. M., Collman, F. & Johnson, G. R. Label-free prediction of three-dimensional fluorescence images from transmitted-light microscopy. *Nat. Methods* **15**, 917–920 (2018).
12. Kandel, M. E. *et al.* Phase imaging with computational specificity (PICS) for measuring dry mass changes in sub-cellular compartments. *Nat. Commun.* **11**, 6256 (2020).

13. Schmidt, U., Weigert, M., Broaddus, C. & Myers, G. Cell Detection with Star-Convex Polygons. in *Medical Image Computing and Computer Assisted Intervention – MICCAI 2018* 265–273 (Springer International Publishing, 2018).
14. Weigert, M., Schmidt, U., Haase, R., Sugawara, K. & Myers, G. Star-convex polyhedra for 3D object detection and segmentation in microscopy. in *2020 IEEE Winter Conference on Applications of Computer Vision (WACV)* (IEEE, 2020). doi:10.1109/wacv45572.2020.9093435.
15. Ardila Riveros, J. C. *et al.* Automated optimization of endoderm differentiation on chip. *Lab Chip* **21**, 4685–4695 (2021).
16. Orozco-Fuentes, S. *et al.* Quantification of the morphological characteristics of hESC colonies. *Sci. Rep.* **9**, 17569 (2019).
17. McColloch, A., Rabiei, M., Rabbani, P., Bowling, A. & Cho, M. Correlation between Nuclear Morphology and Adipogenic Differentiation: Application of a Combined Experimental and Computational Modeling Approach. *Sci. Rep.* **9**, 16381 (2019).
18. Heo, S.-J. *et al.* Differentiation alters stem cell nuclear architecture, mechanics, and mechano-sensitivity. *Elife* **5**, (2016).
19. Ankam, S. *et al.* Temporal Changes in Nucleus Morphology, Lamin A/C and Histone Methylation During Nanotopography-Induced Neuronal Differentiation of Stem Cells. *Front Bioeng Biotechnol* **6**, 69 (2018).
20. Cuomo, A. S. E. *et al.* Single-cell RNA-sequencing of differentiating iPS cells reveals dynamic genetic effects on gene expression. *Nat. Commun.* **11**, 810 (2020).
21. Tinevez, J.-Y. *et al.* TrackMate: An open and extensible platform for single-particle tracking. *Methods* **115**, 80–90 (2017).
22. D'Amour, K. A. *et al.* Efficient differentiation of human embryonic stem cells to definitive endoderm. *Nat. Biotechnol.* **23**, 1534–1541 (2005).
23. Li, Q. *et al.* A sequential EMT-MET mechanism drives the differentiation of human embryonic stem cells towards hepatocytes. *Nat. Commun.* **8**, 15166 (2017).
24. Tauran, Y. *et al.* Analysis of the transcription factors and their regulatory roles during a

step-by-step differentiation of induced pluripotent stem cells into hepatocyte-like cells. *Mol Omics* **15**, 383–398 (2019).

25. Unger, M. A., Chou, H. P., Thorsen, T., Scherer, A. & Quake, S. R. Monolithic microfabricated valves and pumps by multilayer soft lithography. *Science* **288**, 113–116 (2000).
26. Hohwieler, M. *et al.* Human pluripotent stem cell-derived acinar/ductal organoids generate human pancreas upon orthotopic transplantation and allow disease modelling. *Gut* **66**, 473–486 (2017).
27. Shahryari, A. *et al.* Increasing Gene Editing Efficiency for CRISPR-Cas9 by Small RNAs in Pluripotent Stem Cells. *CRISPR J* **4**, 491–501 (2021).
28. Zhu, Z. *et al.* Genome Editing of Lineage Determinants in Human Pluripotent Stem Cells Reveals Mechanisms of Pancreatic Development and Diabetes. *Cell Stem Cell* **18**, 755–768 (2016).
29. Ronneberger, O., Fischer, P. & Brox, T. U-Net: Convolutional Networks for Biomedical Image Segmentation. *arXiv [cs.CV]* (2015).
30. Waibel, D. J. E., Shetab Boushehri, S. & Marr, C. InstantDL: an easy-to-use deep learning pipeline for image segmentation and classification. *BMC Bioinformatics* **22**, 103 (2021).
31. Varoquaux, G. *et al.* Scikit-learn. *GetMob. Mob. Comput. Commun.* **19**, 29–33 (2015).
32. Bernard, J. Python Data Analysis with pandas. in *Python Recipes Handbook: A Problem-Solution Approach* (ed. Bernard, J.) 37–48 (Apress, 2016).

Supplementary Table 1. Chemical compounds used for DE differentiation.

Chemicals	Company, Cat#
MCDB131	Gibco®, Cat# 10372-019
Glutamax	Life Technologies, Cat# 35050-079
BSA	Sigma, Cat# 10775835001
Sodium bicarbonate	Sigma, MO, Cat# S6297
Glucose	Sigma, Cat# G8769
ITS-X	Life Technologies, Cat# 51500-056
Ascorbic acid	Sigma-Aldrich, Cat# A4544
Penicillin/Streptomycin	Gibco®, Cat# 15140-122

Supplements	Company, Cat#
Activin A	Peprotech, Cat# 120-14-300
CHIR-99021	GSK3β inhibitor, Tebu-bio, Cat# 24804-0004
ROCK inhibitor	Y-27632, Santa Cruz Biotechnology, Cat# sc-281642A

Supplementary Table 2. Basal media composition and supplements used for DE differentiations

Basal media
MCDB131
1x Glutamax
0.5% BSA
1.5g/l Sodium Bicarbonate
1.8g/l Glucose
1% penicillin/streptomycin

Component	Day	Concentration
ActA	Day01	100ng/ml
CHIR		5µM
ROCK inhibitor		0µM
ActA	Day02	100ng/ml
CHIR		0.3µM
ActA	Day03	100ng/ml
CHIR		0µM

Supplementary Table 3. Antibodies used in immunochemistry

Antibody	Host	Reference	Fluorophore
anti-OCT4	Mouse IgG1	Cell Signaling 75463	N/A
anti-FOXA2	Rabbit IgG	Cell Signaling 8186S	N/A
anti-SOX17	Goat IgG	Acris/Novus GT15094	N/A
anti-N-Cadherin	Mouse IgG1	BD 610920	N/A
anti-E-Cadherin	Rabbit IgG	Cell Signaling 3195S	N/A
anti-Mouse	Donkey	Invitrogen A21202	Alexa Fluor 488
anti-Rabbit	Donkey	Invitrogen A31572	Alexa Fluor 555
anti-Goat	Donkey	Invitrogen A21082	Alexa Fluor 633
anti-Mouse	Donkey	Dianova 107-175-151	Alexa Fluor 647
anti-Goat	Donkey	Invitrogen A11055	Alexa Fluor 488