Search for temporal cell segmentation robustness in phase-contrast microscopy videos

Estibaliz Gómez-de-Mariscal[1,2] ESGOMEZM@PA.UC3M.ES
[1] *Bioengineering and Aerospace Engineering Department, Universidad Carlos III de Madrid, 28911 Leganés, Spain*
[2] *Instituto de Investigación Sanitaria Gregorio Marañón, 28007 Madrid, Spain*
Hasini Jayatilaka[3]
[3] *AtlasXomics Inc. New Haven, CT, 06511, USA*
Özgün Çiçek[4]
[4] *Department of Computer Science, Albert-Ludwigs-University, Freiburg 79110, Germany*
Thomas Brox[4]
Denis Wirtz [5,6]
[5] *Department of Chemical and Biomolecular Engineering, Institute for Nanobiotechnology, The Johns Hopkins University, Baltimore, Maryland 21218, USA*
[6] *Department of Oncology, The Johns Hopkins University School of Medicine, Baltimore, Maryland 21205, USA*
Arrate Muñoz-Barrutia[1,2] MAMUNOZ@ING.UC3M.ES

Editors: Under Review for MIDL 2022

Abstract

Studying cell morphology changes with time is critical to understanding cell migration mechanisms. This work presents a deep learning-based workflow to segment cancer cells embedded in 3D collagen matrices and imaged with phase-contrast microscopy. Our approach uses transfer learning and recurrent convolutional long-short term memory units to exploit the temporal information from the past and provide. Besides, we propose a geometrical-characterization approach to studying cancer cell morphology. Our approach offers stable results in time, and it is robust to the different weight initialization or training data sampling. We introduce a new annotated dataset for 2D cell segmentation and tracking and an open-source implementation to replicate the experiments or adapt them to novel image processing problems.

Keywords: Video segmentation, cell segmentation, transfer-learning, convlstm, phase-contrast, cell migration, mesenchymal migration

1. Introduction

Metastasis, the leading cause of death caused by cancer, refers to the process in which cells spread from the primary tumor to adjacent tissues, proliferate and invade healthy organs. Thus, understanding the mechanisms driving cancer cell migration is critical to characterize highly metastatic cells, develop new efficient treatments and improve precision medicine. Due to the experimental complexity (e.g., mechanical control of 3D gel matrices, reduced imaging speed and increased phototoxicity), most quantitative cell migration studies are performed on 2D cell culture experiments, while cell migration naturally occurs in 3D environments (i.e., the extracellular matrix (ECM)). Instead of the lamellipodia or filopodia

usually observed in 2D substrates, cells migrating in 3D substrates (*e.g.*, collagen type I matrices) form dendritic protrusions to anchor, exert forces, and propel. Consequently, there is a growing interest to study cell morphology and motility in 3D environments (Doyle et al., 2015; Wu et al., 2018; Jayatilaka et al., 2018; Doyle et al., 2021).

Sample phototoxicity and photobleaching are well-known constraints in fluorescence microscopy. Thus, prolonged time-lapse videos are customarily acquired using brightfield microscopy techniques such as phase contrast at the expense of low contrast between the cell and the background. To mimic the mechanical properties of the ECM, cells are commonly embedded in matrices of collagen type I (Wu et al., 2018; Doyle et al., 2015). Additionally, the brightfield images of collagen matrices are full of artifacts and quite heterogeneous due to the polymerization of the collagen fibers resulting in images full of artifacts (see Figure 3 in the Appendix). The typical setup to study 3D cell motility consists of acquiring images of a fixed focal plane placed in the middle of the collagen matrix where the cells are embedded. Therefore, the cell movement is assessed in the cited xy-plane under the premise that 3D cell motility is isotropic (Wu et al., 2018). Note that while the images are 2D, cells are migrating in 3D, so they can exit and enter the plane of focus, fixed along with the videos.

The manual annotation of cells in long phase-contrast microscopy time-lapse videos is tedious and non-viable. Hence, implementing a robust computational tool for the automatic quantification of cell morpho-dynamics is critical to deciphering the mechanisms that drive cancer cell migration in metastasis.

Deep learning (DL) approaches are considered the state-of-the-art for processing images with large inter-variability and intra-variability. Usiigaci *et al.* (Tsai et al., 2019) released one of the first methods (based on a Mask R-CNN (He et al., 2017) backbone) that achieve accurate cell instance segmentation in phase-contrast microscopy images. Lux and Matula (Lux and Matula, 2020) use a DL-based approach that first predicts the binary segmentation and landmarks for cell detection and then performs a watershed transformation using the landmarks to provide the instance segmentation. Pixel embedding approaches (Payer et al., 2018; Lalit et al., 2021) are efficient instance segmentation strategies, especially for highly packed cells. Other works show accurate results for binary segmentation using recurrences such as the convolutional Long-Short Term Memory (LSTM) U-Net (Arbelle and Raviv, 2019; Wang et al., 2019b) or the recurrent U-Net (Wang et al., 2019a). Additionally, DeepCellKiosk (Bannon et al., 2021) is a cloud-based toolbox to train and deploy DL models for cell segmentation, detection and tracking and to ease the use of large datasets. Some of the previously mentioned methods (*e.g.*, Mask R-CNN, convolutional LSTM U-Net, cosine-embedding) have high-computational requirements when adapting their implementations to new experiments or data types. Others are not optimized for a low cell/background ratio (*e.g.* DeepCellKiosk models for instance segmentation, the approach of Filip ***et al.***). Additionally, most previous contributions focus on defining the convolutional neural network (CNN) architecture and, sometimes, the loss function rather than the training strategy. Aspects to consider for the latter are using pre-trained encoders, dealing with data imbalance, or avoiding the creation of artifacts during data augmentation. From our in-house experiments, we realized that the quality of the ground truth and the training strategy was the most important factors to get accurate and robust results.

We want to introduce the temporal-consistency (Varghese et al., 2020) as we, as manual annotators, needed it to achieve a robust segmentation that can be tracked afterwards. As

stated in (Ulman et al., 2017), cell segmentation and tracking could be achieved in different manners, while the most common one is to perform first, a consistent segmentation in time, and then the tracking. We identify two possible approaches to obtain consistent segmentation: (1) using 3D convolutions (2D images and the time), and (2) using recurrent attributes in the network. We claim that a system able to discard the noise and artifacts and detect the movement of the cell will be sufficient to determine the cellular shape accurately. Cell movement can be easily detected by the shape differences from frame to frame. A similar paradigm for time consistency can be found in echocardiography segmentation due to the cardiac function (Wei et al., 2020; Painchaud et al., 2021). The latter works proposed exploiting 3D-connectivity of a 3D U-Net to extract time information. Nevertheless, 3D CNN implementations require considerable memory while the cell movement can be easily detected by the local shape differences between frames. Hence, inspired by the work of Arbelle *et al.* (Arbelle and Raviv, 2019), we decided to build a recurrent architecture that combines a 2D U-Net shaped encoder-decoder with a convolutional long short term memory (ConvLSTM) (architecture, Figure 5 and convolutional blocks, Figure 6 in the Appendix).

We need to detect cell protrusions to analyze the relationship between cell morphology and motility. There exist several works in the literature addressing a closely related problem with cell filopodium (Maška et al., 2013; Tsygankov et al., 2014; Barry et al., 2015; Urbančič et al., 2017; Castilla et al., 2019; Bagonis et al., 2019). The previously cited methods work under the hypothesis that a well-defined frontier exists between the cell and the filopodia. However, setting the limit between the cell body and the cell protrusions is still an open question. This inconvenience can be circumvented by detecting the protrusion tips instead.

To summarize, in this work, we propose deep learning (DL)-based bioimage processing workflow: (1) to segment cells on phase-contrast microscopy videos, and (2) to detect their protrusions. We first build a consistent and heterogeneous ground truth image dataset. Then, we propose a workflow that combines deep CNN and geometrical analysis of the cell morphology to quantify cellular protrusions automatically.

2. Materials

We build the ground truth dataset from the phase-contrast microscopy videos used in (Jayatilaka et al., 2018): Human fibrosarcoma HT1080WT (ATCC) cells at low cell densities embedded in 3D collagen type I matrices. The time-lapse videos were recorded every 2 minutes for 16.7 hours and covered a field of view of 1002 pixels × 1004 pixels with a pixel size of $0.802 \mu m/pixel$ The videos were pre-processed to correct frame-to-frame drift artifacts, resulting in a final size of 983 pixels × 985 pixels. All the details are given in Appendix A.

The ground truth is built with heterogeneous and independent videos. We chose 27 videos from independent replicates in which mitosis and apoptosis events and different cell morphology and migration patterns were present. We ensure that cells touching each other and migrating faster or slower are also present. The variation of the collagen matrix under the microscope is also considered. We subtract short sections between 3 and 100 frames from the videos to gather the mentioned events. Finally, our ground truth data consists of 56 short videos, resulting in a total of 992 frames. Over time, the instance segmentation of focused cells is manually annotated and uniquely labeled, preserving the tracking information. Three experts annotated the videos, and a majority voting method as described in (Ulman et al.,

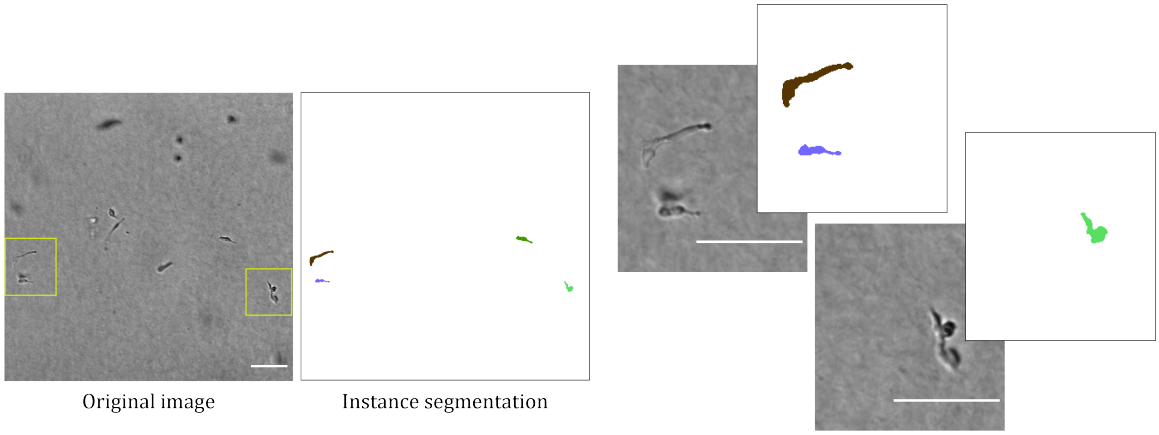

Figure 1: Sample image of a human fibrosarcoma (HT1080WT) cells embedded in a 3D collagen Type I matrix, From left to right: Phase contrast microscopy image; manual annotations of cells in focus in (a); Zoomed versions of the crops shown in the yellow boxes in (a). Cells out of focus are discarded. Scale bars of 100μm.

2017) was applied to combine the three annotations and build a consensus ground truth. The ground truth data can be accessed in https://zenodo.org/record/5979761.

3. Cell segmentation

3.1. Methodology

We propose to train a U-Net-shaped encoder-decoder with a pre-trained encoder, depth-wise separable convolutions, and convolutional long short term memory (LSTM) (Shi et al., 2015) units to obtain a binary mask with the cells being represented by the foreground pixels.

In (Arbelle and Raviv, 2019), the authors add a ConvLSTM on each encoder level, so the temporal and spatial information are encoded together. Using a pre-trained encoder, the ConvLSTM units should go in the decoder layers. We use a single ConvLSTM layer at the end of the encoder-decoder to optimize the memory usage and adapt to the limitations of our hardware training systems. Thus, the number of parameters is not significantly increased despite the recurrent layers. Separable depth-wise convolutions are also introduced to optimize the memory usage, as proposed in (Howard et al., 2017) (see Figure 5 in the Appendix). The last recurrent layer will sequentially analyze the frames of an input video once the 2D U-Net processes them. With this approach, we expect to improve the robustness to intra-cell variations along time (i.e., caused by cell exiting and entering the plane of focus) and the network's output to be more consistent at the cell edges where the protrusions appear and disappear (see Figure 4 in the Appendix). The proposed architecture provides segmentation for the frame at time t based on the information in the frames $(t-k,...,t)$ for

a chosen time window k.
The architecture is as follows:

- **2D U-Net shaped encoder-decoder**: The encoder is a MobileNetV2 (Sandler et al., 2018) pre-trained on the ImageNet (Deng et al., 2009) dataset for image classification with 35% of the filters of its convolutional layers and with skip connections. A decoder with separable depth-wise convolutions is connected to the skip connections of the MobileNetV2.
- **A recurrent Conv2D-LSTM layer:** The time-series entering the encoder-decoder will be recurrently processed so the segmentation of the frame t is done taking into account previous information.
- **A final 2D convolutional layer** with two feature maps to obtain the pixel classification (background and foreground).

Training data sampling strategy: A common practice to apply data augmentation (DA) is to crop the original images in a large number of patches and modify these patches with random image transformations. Some DA transformations use image mirroring or zero-padding in the image borders, which adds many unrealistic artifacts that would prevent our CNNs from learning correctly. Frequently, the patches are cropped uniformly along the image regardless of the foreground-background ratio. Our proposal consists of first transforming the image and then cropping a patch using a probability distribution function that deals with the foreground-background ratio. The probability distribution function is commonly used in statistical learning when there is data imbalance. In our case, we define the pixel probability distribution function by setting a weight of 50000 and 1 to the pixels in the foreground and background, respectively. Note that the ratio of foreground to background is a maximum of 0.06 in the best case. After setting the weights, the image is normalized. Namely, the sum of all the pixels equals 1, which means we create a probability distribution function for all the pixels in the image. A random pixel drawn using this probability distribution is set as the centroid of the patch that will enter the network during training. With this configuration, we generate cell-containing patches most of the time, and just in a few cases, background patches. The previous method has been implemented in Python using TensorFlow. The code is freely available at https://github.com/esgomezm/microscopy-dl-suite-tf.

3.2. Experimental results

A set of 28 patches from the raw data in the training set were used as a validation set. Those patches remain the same for all our experiments, while the training data is randomly transformed on each iteration. Each input image intensity is normalized using the percentile normalization with 0.1 and 99.1 percentages for the lower and upper bounds. The models are trained in two steps:

1. Transfer-learning to our segmentation model of the MobileNetV2 weights for image classification. In this step, the pre-trained encoder is frozen, the decoder is randomly

initialized, and the model is trained for $2K$ epochs. This model (i.e., the decoder) is trained to learn the segmentation task using a large learning rate (0.005). The proposed transfer-learning approach is not sensible to the decoder initialization and the data sampling strategy (see Appendix).

2. We unfreeze the encoder and fine-tune all the weights using a lower initial learning rate (0.0001) that is automatically halved whenever the Jaccard index does not show an improvement larger than 0.0001 during 600 epochs in the validation set. We train for $2K$ epochs (see Table 1, and Figures 2 and 10).

We evaluate five different configurations of the architecture Bi, $i = 1, ..., 5$ by changing the depth and sizes of the convolutional layers in the decoder. The main architectural details are given in Table 1. Notice that the number of trainable parameters prevents us from testing a batch size larger than two using our computational resources (see the Appendix). The decoders of the CNN setups in Table 1 are randomly initialized using the Glorot Uniform initializer (Glorot and Bengio, 2010). We choose exponential linear units (ELUs) to activate the convolutional layers in the decoder except for the output layers, which are not activated. We apply the categorical cross-entropy loss function given by Equation 1 directly to the logits of the last layer. The latter is recommended for a more numerically stable computation of the gradients[1]. ADAM is the optimizer chosen (Equation 2). During the training, we evaluate the binary segmentation with the Jaccard index, Equation 3. The accuracy is assessed on the test set. The binary segmentation is evaluated using the Evaluation Software provided in the Cell Tracking Challenge (Ulman et al., 2017) (SEG measure). The Appendix describes the total SEG that accounts for all the videos in the test set.

	Pools	Convolutional filters	Batch Size	Transfer learning	Fine tuning
B1	5	$16-32-64-128-256$	1	0.473	0.430
B2	4	$25-50-100-200$	1	0.411	0.455
B3	4	$25-50-100-200$	2	0.550	0.551
B4	4	$50-100-200-400$	1	0.481	0.446
B5	3	$16-32-64$	2	0.437	0.561

Table 1: Convolutional neural network architecture and SEG for transfer learning and fine tuning. Note that the size of the convolutional layers only applies for the decoder path of the encoder-decoder. See Figure 5 in the Appendix for details about the architecture. Transfer learning and fine tuning processes are programmed for $2K$ epochs with a constant learning rate of 0.005 and 0.0001, respectively.

Unsurprisingly, a batch size larger than one improves the learning process making it smoother and more accurate. While the loss function for the training and validation datasets differs in shape and magnitude, the accuracy measured by the Jaccard index (Equation 3) remains similar for both datasets, which could be due to the overlap between the training

1. https://www.tensorflow.org/api_docs/python/tf/keras/losses/SparseCategoricalCrossentropy

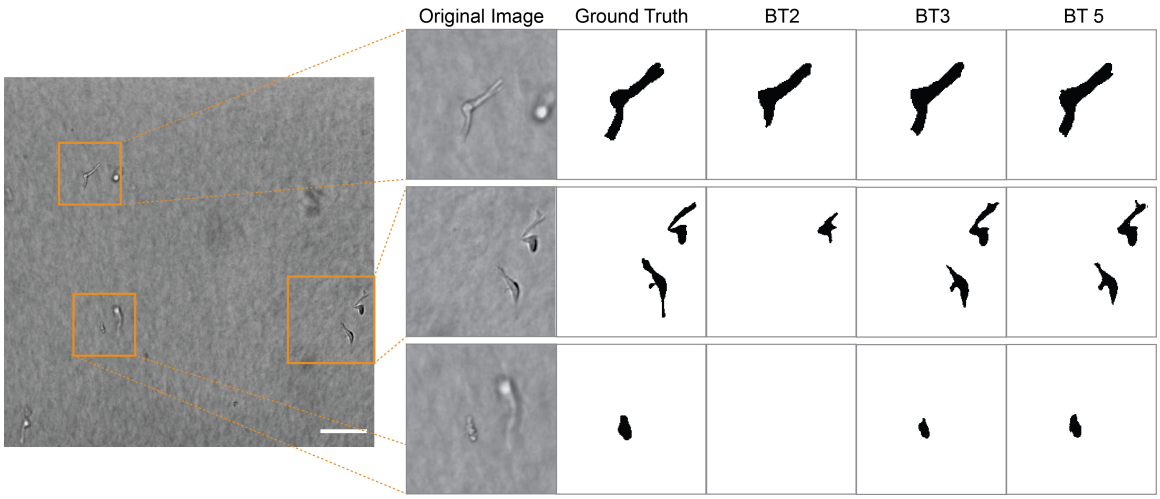

Figure 2: (Left) Phase-contrast microscopy image. Scale bar of 100 μm; (Right) Zoomed crops of the region enclosed by the boxes in the full images; (From left to right) Original phase-contrast microscopy crop; Ground Truth (annotated in-focus cells are labeled in black); Output of the trained models B2, B3 and B5.

and validation sets. The CNN set up for B4 is the least stable, related to the larger size of the convolutional layers chosen and the small batch size. Table 1 shows that B3 and B5 are the most accurate CNNs.

We trained a model with the same architecture as B5 but analysed only one input image, thus missing the time information. When comparing both (Figure 11 in the Appendix), one can notice that integrating the temporal information improves the segmentation result.

4. Protrusion tip quantification

Similar to the work of (Castilla et al., 2019), we use the cell's skeleton and its end-points to identify the tips of the protrusions. When annotated manually, biologists consider cell protrusions that are longer than 5 μm. We use the Geodesic distance transform to measure the distance between the cell centroid and each detected tip and estimate the protrusion length. Hence, spurious tips can be accordingly discarded using this longitudinal estimation. We set the minimum length to 20 μm. Note that the Geodesic distance transform includes the radius of the cell nuclei region computed from the cell body centroid. Most cells show a roundish pattern in their nucleic region. As cells are either elongated or rounded, the nuclei area will always be the wider region of the cell. Hence, the centroid is determined by the location of the maximum value of the Euclidean distance transform (see Figures 12 and 13).

5. Discussion and conclusions

The results obtained when using pre-trained encoders are already quite promising in terms of true and false positive pixel classification, suggesting that pre-trained encoders on biomedical images could accelerate and improve the learning process. However, the outcome of the fine-tuned models improved, especially for those training schedules with batch sizes larger than one (see Tables 1). Unfortunately, the batch sizes we could use were limited by the resources to train the CNN (see the number of parameters for each configuration in the Appendix). We use depth-wise separable convolutions to increase the field of view of the network at a lower parameter cost. Nevertheless, the receptive field of a pixel in our networks varies from 194 pixels $\times$ 194 pixels to 230 pixels $\times$ 230 pixels, depending on the encoder-decoder depth chosen. Hence, reducing the spatial input size to allow larger batch sizes can prevent the network from visualizing the entire cell body without being affected by the padding artifacts of the convolutional layers.

Another critical point in the image processing design is the trade-off between image resolution and network configuration. The thinnest details, i.e., the cell protrusions, are the primary source of error for the segmentation. We attribute these errors to the poor image resolution to draw cell protrusions and their large variability. Such low resolution hinders the learning of essential features and voids the effect that those pixels may have in the loss function or gradients. A simple computational experiment to test it could be to upsample the size of the images and repeat the training process.

Training strategy plays a crucial role in DL model training with scarce annotated data. The proposed training data sampling is relatively easy to implement and resembles the approach described in (Ronneberger et al., 2015). It can hardly worsen the final results and often achieves similar performances with simpler CNN architectures. We believe that most image processing DL workflows should integrate a similar arrangement.

Thanks to the binary cell segmentation, we obtain accurate results for the cell tracking in low-resolution 2D phase-contrast microscopy images. Due to the low density of cells in our images, they could be tracked easily with the last version of TrackMate (Tinevez et al., 2017; Ershov et al., 2021) which already processes object instances. The combination of whole-cell shape segmentation and tracking is already a promising image processing tool for studying new clinical strategies to mitigate metastatic phenotypes.

Acknowledgments

This work was partially funded by Ministerio de Ciencia, Innovación y Universidades, Agencia Estatal de Investigación, under Grant PID2019-109820RB-I00, MCIN / AEI / 10.13039/501100011033/, co-financed by European Regional Development Fund (ERDF), "A way of making Europe" (AMB); BBVA Foundation under a 2017 Leonardo Grant for Researchers and Cultural Creators (AMB); the US National Institutes of Health under Grants UO1AG060903 (DW) and U54CA143868 (DW). We also want to acknowledge the support of NVIDIA Corporation with the donation of the Titan X (Pascal) GPU used for this research.

References

Assaf Arbelle and Tammy Riklin Raviv. Microscopy Cell Segmentation Via Convolutional LSTM Networks. In *2019 IEEE 16th International Symposium on Biomedical Imaging (ISBI 2019)*, pages 1008–1012, 2019. doi: 10.1109/ISBI.2019.8759447.

Maria M. Bagonis, Ludovico Fusco, Olivier Pertz, and Gaudenz Danuser. Automated profiling of growth cone heterogeneity defines relations between morphology and motility. *J. Cell Biol.*, 218(1):350–379, jan 2019. ISSN 0021-9525. doi: 10.1083/jcb.201711023.

Dylan Bannon, Erick Moen, Morgan Schwartz, Enrico Borba, Takamasa Kudo, Noah Greenwald, Vibha Vijayakumar, Brian Chang, Edward Pao, Erik Osterman, et al. Deepcell kiosk: scaling deep learning–enabled cellular image analysis with kubernetes. *Nature Methods*, 18(1):43–45, 2021. doi: 10.1038/s41592-020-01023-0.

David J. Barry, Charlotte H. Durkin, Jasmine V. Abella, and Michael Way. Open source software for quantification of cell migration, protrusions, and fluorescence intensities. *J. Cell Biol.*, 209(1):163–180, apr 2015. ISSN 0021-9525. doi: 10.1083/jcb.201501081.

Carlos Castilla, Martin Maska, Dmitry V. Sorokin, Erik Meijering, and Carlos Ortiz-de Solorzano. 3-D Quantification of Filopodia in Motile Cancer Cells. *IEEE Trans. Med. Imaging*, 38(3):862–872, mar 2019. ISSN 0278-0062. doi: 10.1109/TMI.2018.2873842.

Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*, pages 248–255. Ieee, 2009. doi: 10.1109/CVPR.2009.5206848.

Andrew D Doyle, Nicole Carvajal, Albert Jin, Kazue Matsumoto, and Kenneth M Yamada. Local 3D matrix microenvironment regulates cell migration through spatiotemporal dynamics of contractility-dependent adhesions. *Nature Communications*, 6(1):1–15, 2015. doi: 10.1038/ncomms9720.

Andrew D Doyle, Daniel J Sykora, Gustavo G Pacheco, Matthew L Kutys, and Kenneth M Yamada. 3D mesenchymal cell migration is driven by anterior cellular contraction that generates an extracellular matrix prestrain. *Developmental Cell*, 56(6):826–841, 2021. doi: 10.1016/j.devcel.2021.02.017.

Dmitry Ershov, Minh-Son Phan, Joanna W. Pylvänäinen, Stéphane U. Rigaud, Laure Le Blanc, Arthur Charles-Orszag, James R. W. Conway, Romain F. Laine, Nathan H. Roy, Daria Bonazzi, Guillaume Duménil, Guillaume Jacquemet, and Jean-Yves Tinevez. Bringing TrackMate in the era of machine-learning and deep-learning. *bioRxiv*, 2021. doi: 10.1101/2021.09.03.458852.

Xavier Glorot and Yoshua Bengio. Understanding the difficulty of training deep feedforward neural networks. In *Proceedings of the thirteenth international conference on artificial intelligence and statistics*, pages 249–256. JMLR Workshop and Conference Proceedings, 2010.

Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross Girshick. Mask R-CNN. In *Proceedings of the IEEE international conference on computer vision*, pages 2961–2969, 2017.

Andrew G Howard, Menglong Zhu, Bo Chen, Dmitry Kalenichenko, Weijun Wang, Tobias Weyand, Marco Andreetto, and Hartwig Adam. MobileNets: Efficient convolutional neural networks for mobile vision applications. *arXiv preprint arXiv:1704.04861*, 2017.

Hasini Jayatilaka, Anjil Giri, Michelle Karl, Ivie Aifuwa, Nicholaus J Trenton, Jude M Phillip, Shyam Khatau, and Denis Wirtz. EB1 and cytoplasmic dynein mediate protrusion dynamics for efficient 3-dimensional cell migration. *FASEB J.*, 32(3):1207–1221, mar 2018. ISSN 0892-6638. doi: 10.1096/fj.201700444RR.

Manan Lalit, Pavel Tomancak, and Florian Jug. Embedding-based instance segmentation of microscopy images. *arXiv preprint arXiv:2101.10033*, 2021.

Filip Lux and Petr Matula. Cell segmentation by combining marker-controlled watershed and deep learning. *arXiv preprint arXiv:2004.01607*, 2020.

Martin Maška, Xabier Morales, Arrate Muñoz-Barrutia, Ana Rouzaut, and Carlos Ortiz-de Solórzano. Automatic quantification of filopodia-based cell migration. In *2013 IEEE 10th International Symposium on Biomedical Imaging*, pages 668–671. IEEE, 2013. doi: 10.1109/ISBI.2013.6556563.

Pavel Matula, Martin Maška, Dmitry V. Sorokin, Petr Matula, Carlos Ortiz-de Solórzano, and Michal Kozubek. Cell Tracking Accuracy Measurement Based on Comparison of Acyclic Oriented Graphs. *PLOS ONE*, 10(12):e0144959, dec 2015. ISSN 1932-6203. doi: 10.1371/journal.pone.0144959.

Nathan Painchaud, Nicolas Duchateau, Olivier Bernard, and Pierre-Marc Jodoin. Echocardiography segmentation with enforced temporal consistency. *arXiv preprint arXiv:2112.02102*, 2021.

Christian Payer, Darko Štern, Thomas Neff, Horst Bischof, and Martin Urschler. Instance segmentation and tracking with cosine embeddings and recurrent hourglass networks. In *International Conference on Medical Image Computing and Computer-Assisted Intervention*, pages 3–11. Springer, 2018. doi: 10.1016/j.media.2019.06.015.

Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-Net: Convolutional Networks for Biomedical Image Segmentation. In Springer, editor, *Int. Conf. Med. image Comput. Comput. Interv.*, pages 234–241. Springer International Publishing, 2015. ISBN 9783319245737. doi: 10.1007/978-3-319-24574-4_28.

Curtis T. Rueden, Johannes Schindelin, Mark C. Hiner, Barry E. DeZonia, Alison E. Walter, Ellen T. Arena, and Kevin W. Eliceiri. ImageJ2: ImageJ for the next generation of scientific image data. *BMC Bioinformatics*, 18(1):529, dec 2017. ISSN 1471-2105. doi: 10.1186/s12859-017-1934-z.

Mark Sandler, Andrew Howard, Menglong Zhu, Andrey Zhmoginov, and Liang-Chieh Chen. MobileNetV2: Inverted residuals and linear bottlenecks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4510–4520, 2018.

Johannes Schindelin, Ignacio Arganda-Carreras, Erwin Frise, Verena Kaynig, Mark Longair, Tobias Pietzsch, Stephan Preibisch, Curtis Rueden, Stephan Saalfeld, Benjamin Schmid, Jean-Yves Tinevez, Daniel James White, Volker Hartenstein, Kevin Eliceiri, Pavel Tomancak, and Albert Cardona. Fiji: an open-source platform for biological-image analysis. *Nat. Methods*, 9(7):676–682, jul 2012. ISSN 1548-7091. doi: 10.1038/nmeth.2019.

Caroline A Schneider, Wayne S Rasband, and Kevin W Eliceiri. NIH Image to ImageJ: 25 years of image analysis. *Nat. Methods*, 9(7):671–675, jul 2012. ISSN 1548-7091. doi: 10.1038/nmeth.2089.

Alexandra B. Schroeder, Ellen T. A. Dobson, Curtis T. Rueden, Pavel Tomancak, Florian Jug, and Kevin W. Eliceiri. The ImageJ ecosystem: Open-source software for image visualization, processing, and analysis. *Protein Sci.*, 30(1):234–249, jan 2021. ISSN 0961-8368. doi: 10.1002/pro.3993.

George Seif. Semantic segmentation suite in tensorflow. https://github.com/GeorgeSeif/Semantic-Segmentation-Suite, 2018.

Xingjian Shi, Zhourong Chen, Hao Wang, Dit-Yan Yeung, Wai kin Wong, and Wang chun Woo. Convolutional LSTM Network: A Machine Learning Approach for Precipitation Nowcasting, 2015.

Jean-Yves Tinevez, Nick Perry, Johannes Schindelin, Genevieve M Hoopes, Gregory D Reynolds, Emmanuel Laplantine, Sebastian Y Bednarek, Spencer L Shorte, and Kevin W Eliceiri. TrackMate: An open and extensible platform for single-particle tracking. *Methods*, 115:80–90, 2017. doi: 10.1016/j.ymeth.2016.09.016.

Hsieh-Fu Tsai, Joanna Gajda, Tyler F.W. Sloan, Andrei Rares, and Amy Q. Shen. Usiigaci: Instance-aware cell tracking in stain-free phase contrast microscopy enabled by machine learning. *SoftwareX*, 9:230–237, jan 2019. ISSN 23527110. doi: 10.1016/j.softx.2019.02.007.

Denis Tsygankov, Colleen G. Bilancia, Eric A. Vitriol, Klaus M. Hahn, Mark Peifer, and Timothy C. Elston. CellGeo: A computational platform for the analysis of shape changes in cells with complex geometries. *J. Cell Biol.*, 204(3):443–460, feb 2014. ISSN 1540-8140. doi: 10.1083/jcb.201306067.

Vladimír Ulman, Martin Maška, Klas E G Magnusson, Olaf Ronneberger, Carsten Haubold, Nathalie Harder, Pavel Matula, Petr Matula, David Svoboda, Miroslav Radojevic, Ihor Smal, Karl Rohr, Joakim Jaldén, Helen M. Blau, Oleh Dzyubachyk, Boudewijn Lelieveldt, Pengdong Xiao, Yuexiang Li, Siu-Yeung Cho, Alexandre C. Dufour, Jean-Christophe Olivo-Marin, Constantino C. Reyes-Aldasoro, Jose A. Solis-Lemus, Robert Bensch, Thomas Brox, Johannes Stegmaier, Ralf Mikut, Steffen Wolf, Fred A. Hamprecht, Tiago Esteves, Pedro Quelhas, Ömer Demirel, Lars Malmström, Florian Jug, Pavel Tomancak, Erik Meijering, Arrate Muñoz-Barrutia, Michal Kozubek, and Carlos Ortiz-de Solorzano. An objective comparison of cell-tracking algorithms. *Nat. Methods*, 14(12):1141–1152, dec 2017. ISSN 1548-7091. doi: 10.1038/nmeth.4473.

Vasja Urbančič, Richard Butler, Benjamin Richier, Manuel Peter, Julia Mason, Frederick J Livesey, Christine E Holt, and Jennifer L Gallop. Filopodyan: An open-source pipeline for the analysis of filopodia. *J. Cell Biol.*, 216(10):3405–3422, oct 2017. ISSN 0021-9525. doi: 10.1083/jcb.201705113.

Serin Varghese, Yasin Bayzidi, Andreas Bar, Nikhil Kapoor, Sounak Lahiri, Jan David Schneider, Nico M Schmidt, Peter Schlicht, Fabian Huger, and Tim Fingscheidt. Unsupervised temporal consistency metric for video segmentation in highly-automated driving. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops*, pages 336–337, 2020.

Wei Wang, Kaicheng Yu, Joachim Hugonot, Pascal Fua, and Mathieu Salzmann. Recurrent U-Net for resource-constrained segmentation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 2142–2151, 2019a. doi: 10.1109/ICCV.2019.00223.

Yiwen Wang, Ye Lyu, Yanpeng Cao, and Michael Ying Yang. Deep Learning for Semantic Segmentation of UAV Videos. In *IGARSS 2019 - 2019 IEEE International Geoscience and Remote Sensing Symposium*, pages 2459–2462, 2019b. doi: 10.1109/IGARSS.2019.8899786.

Hongrong Wei, Heng Cao, Yiqin Cao, Yongjin Zhou, Wufeng Xue, Dong Ni, and Shuo Li. Temporal-consistent segmentation of echocardiography with co-learning from appearance and shape. In *International Conference on Medical Image Computing and Computer-Assisted Intervention*, pages 623–632. Springer, 2020.

Pei-Hsun Wu, Daniele M Gilkes, and Denis Wirtz. The Biophysics of 3D Cell Migration. *Annu. Rev. Biophys.*, 47(1):549–567, may 2018. ISSN 1936-122X. doi: 10.1146/annurev-biophys-070816-033854.

Appendix A. Microscopy data specifications

Human fibrosarcoma HT1080WT (ATCC) cells at low cell densities were embedded in 3D collagen type I matrices ($5,000-10,000$ cells per $500\,\mu$mL of collagen matrix) and placed on independent plates. They were imaged with a Cascade 1K CCD camera (Roper Scientific) mounted on a Nikon TE2000 microscope with a 10X objective lens (i.e., low magnification). The time-lapse videos were recorded every 2 minutes on a focal plane of at least 200 μm away from the bottom of the culture plates to diminish edge effects. All the videos covered a field of view of 809 $\mu m \times 810$ μm and a total of 16.7 hours (1002 pixels $\times$ 1004 pixels $\times$ 500 frames).

The videos suffer of frame-to-frame drift artifacts due to the microscope objective's drive when acquiring the temporal frames of each well in the plate. Because this feature is present in all the videos, we corrected the drift by applying an affine registration based on the compensation of the image correlation similarity frame-by-frame with the `StackReg` plugin (`http://bigwww.epfl.ch/thevenaz/stackreg/`) for ImageJ (Schneider et al., 2012; Schindelin et al., 2012; Schroeder et al., 2021; Rueden et al., 2017). Then, the videos were cropped to curate border artifact, resulting in a final size of 983 pixels $\times$ 985 pixels $\times$ 500 pixels. See Figure 3 and 4 for sample images.

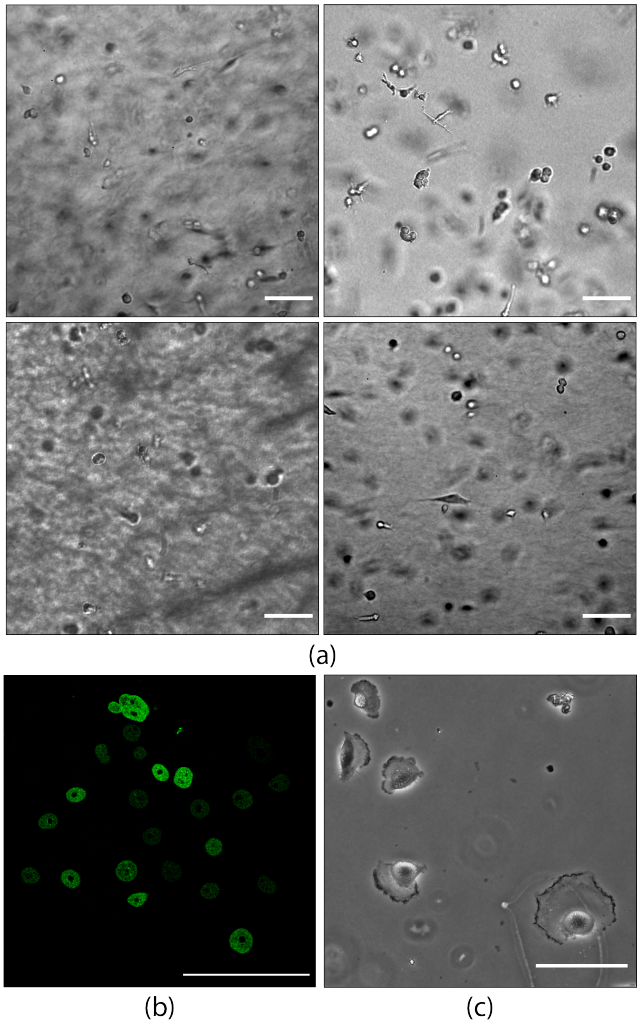

Figure 3: (a) Phase contrast microscopy images of cancer cells (MDA-MB-231) migrating in a 3D collagen type I matrix. The collagen gels, cell culture and acquisition of each image was performed by a different researcher; (b) Fluorescence microscopy image of GFP-GOWT1 mouse stem cells migrating in 2D and fixed in paraformaldehyde (Ulman et al., 2017); (c) Phase contrast microscopy image of glioblastoma-astrocytoma (U373) cells on a polyacrylamide substrate (Ulman et al., 2017). Scale bars of 100 μm in all the images.

Appendix B. Details for the cell segmentation deep learning model

B.1. Convolutional neural network blocks and layers

The main blocks and layer of the recurrent neural network component are given in Figure 6.

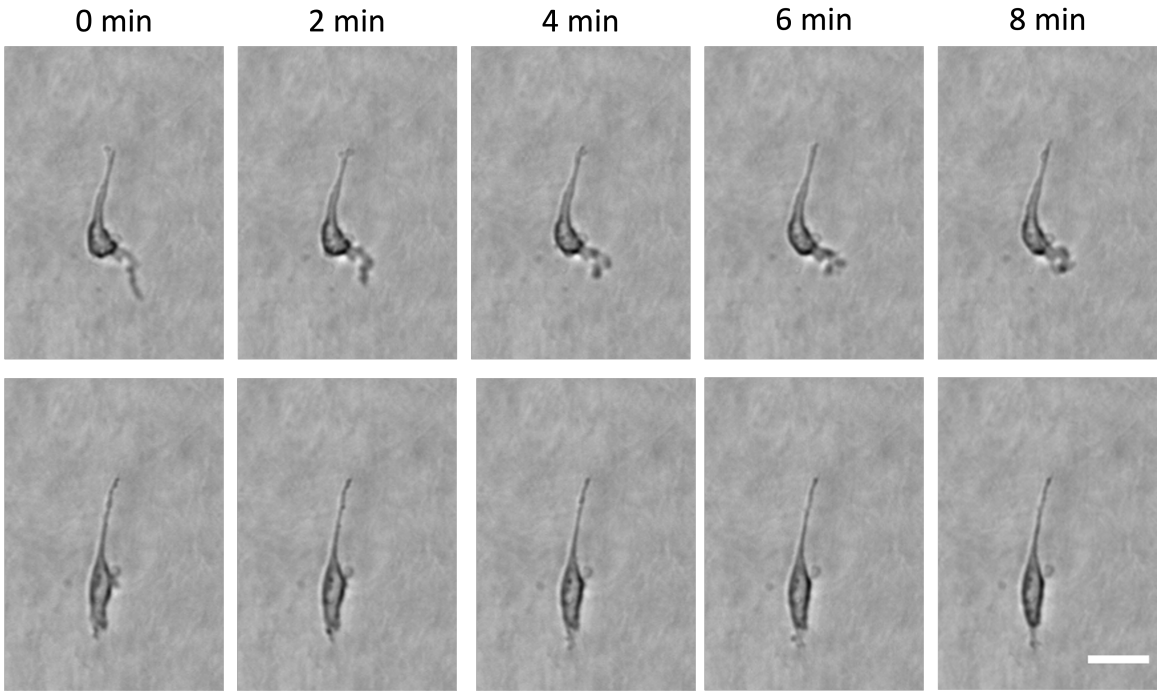

Figure 4: Example of a cell migrating during 8 minutes at two different time points. Cell protrusions lie sometimes out of the focus plane so they are blurry, preventing resolving their correct structure. In the second row, there is an artifact on the right side of the cell that, unless the video is shown for long enough times, it is not possible to determine that it does not belong to the cell body. Scale bars of 5 μm in all the images.

B.2. Loss function and accuracy metrics

The categorical cross-entropy loss function is given as:

$$\mathcal{L}_{cell}(x, y) = -\sum_{i=1}^{C=2} x_i \log(y_i), \tag{1}$$

where C is total number of classes (background and foreground), x_i is the class in the ground truth image and y_i the score given by the CNN.

Adaptive moment estimation (ADAM) is defined as follows:

$$m_t = \beta_1 m_{t-1} + (1 - \beta_1) g_t, v_t = \beta_2 m_{t-1} + (1 - \beta_2) g_t^2, \tag{2}$$

where m_t and v_t are the estimates of the first moment and the second moment of the gradients, respectively, and g_t refers to the gradient value at the time point t. We chose the default values given in Keras: $\beta_1 = 0.9$ and $\beta_2 = 0.999$.

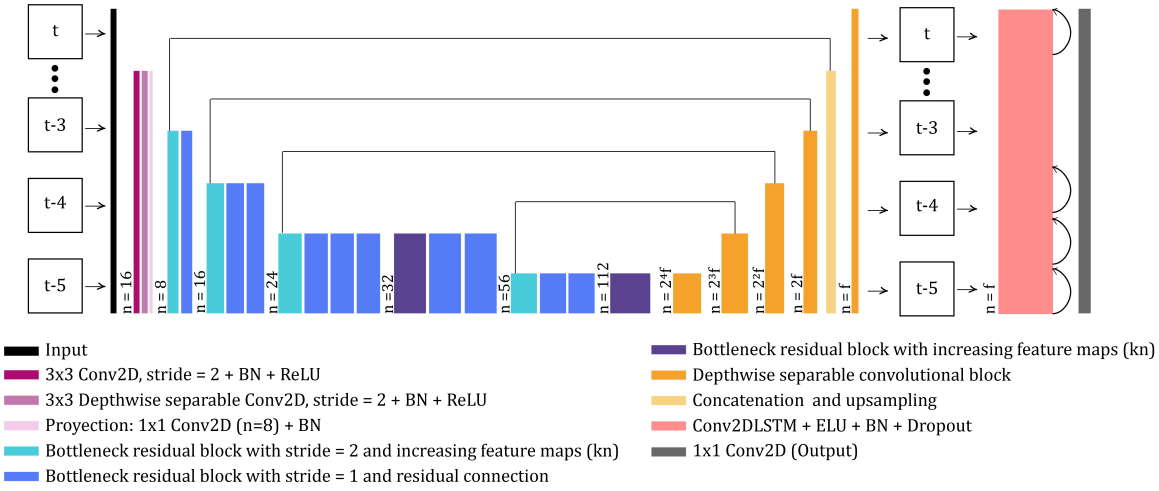

Figure 5: Convolutional neural network (CNN) architecture exploits spatio-temporal information. The architecture processes N consecutive frames ($N = 5$) to infer the segmentation of the last one. The encoder-decoder processes the input frames as a batch. The last layer uses a ConvLSTM to process the time-frames recurrently. The encoder is a pre-trained MobileNetV2 (Sandler et al., 2018) with 35% of its original width (i.e., number of filters) and skip connections to the corresponding blocks in the decoder. In the encoder, the number of feature maps is kn for each $k = 1, 2, 3$ level after downsampling, being $n = 8$. Each block of the decoder is formed by a depth-wise separable convolutional block (see Figure 6). The number of filters in the decoder and the ConvLSTM depend on the parameter f.

The Jaccard index of the foreground is given as follows:

$$JC = \frac{|X \cap Y|}{|X \cup Y|} \tag{3}$$

where X is the ground truth and Y is the output segmentation.

The Evaluation Software provided in the Cell Tracking Challenge (Ulman et al., 2017)[2] calculates segmentation (SEG) and tracking (TRA) accuracy measures. SEG is computed as the Jaccard index between the labeled objects in the ground truth and the resulting masks when the former covers more than 50% of the output mask. Otherwise (less than 50% of overlap), it sets the segmentation measure to zero. This accuracy measure is called SEG. As the length of the test videos is different, we averaged the SEG values of all the videos, Equation 4

$$\overline{SEG}_{test} = \frac{1}{\|N\|} \sum_{i=1}^{N} SEG_i(X_i, Y_i) \tag{4}$$

2. http://celltrackingchallenge.net/evaluation-methodology/

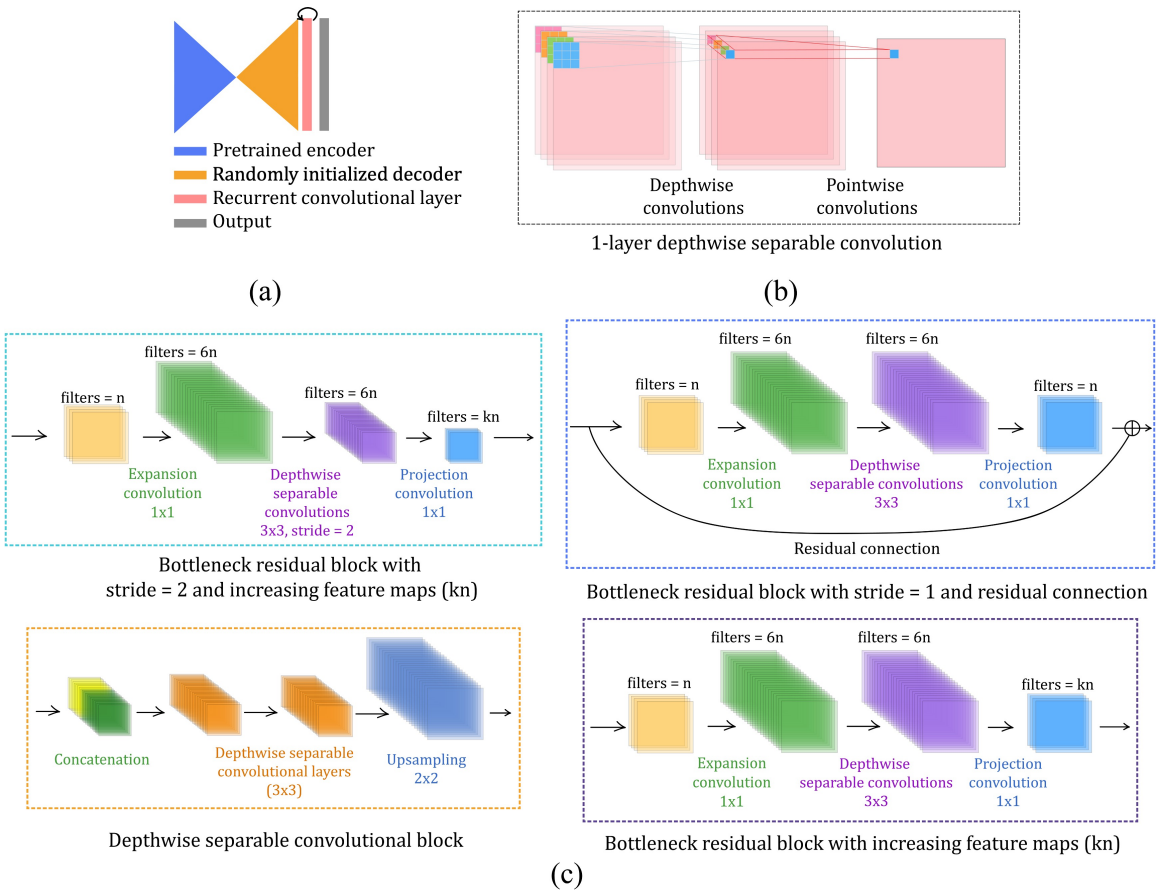

Figure 6: Recurrent convolutional neural network components: a) Single and multi-output modalities of the encoder-decoder described in Figure 5. A MobileNetV2 trained on the ImageNet dataset (Deng et al., 2009) was used as a pre-trained encoder. The decoder and the ConvLSTM units were randomly initialized; b) Description of a depth-wise separable convolution: (1) each channel of the input is filtered independently (depth-wise convolution), and (2) the resulting filtered channels are processed with a kernel of size $1 \times 1 \times c$, c being the number of channels in the input. If the depth-wise separable convolution has n filters, the described process is repeated n times; c) Description of each convolutional block used in the CNN architecture of Figure 5. The expansion and the depth-wise separable convolutions are followed by batch normalization and a ReLU activation function in the encoder path. The proyection convolutions are only followed by a batch normalization layer and are not activated. The depth-wise separable convolutions in the decoder are followed by batch normalization, an ELU activation function, and use dropout.

where X_i and Y_i is video i in the ground truth and the network's output, respectively, and N is the total number of videos in the test set. TRA relies on acyclic oriented graphs matching (AOGM) (Matula et al., 2015), calculated as follows:

$$\mathrm{TRA} = 1 - \frac{\min(\mathrm{AOGM_D}, \mathrm{AOGM_0})}{\mathrm{AOGM_0}} \tag{5}$$

where AOGM_D is the cost of transforming a set of nodes provided by the algorithm into the set of Ground Truth (GT) nodes, and AOGM_0 is the cost of creating the set of GT nodes from scratch (i.e., it is AOGM_D for empty results). TRA behaves as an accuracy measure with values normalized to the $[0, 1]$ interval. The final tracking measure TRA is the average of the TRA values obtained for each video

$$\overline{TRA}_{test} = \frac{1}{\|N\|} \sum_{i=1}^{N} TRA_i(X_i, Y_i). \tag{6}$$

B.3. Number of parameters of the convolutional neural networks

See Table 2.

	Transfer learning	Fine tuning	Total
B1	227,266	477,250	490,530
B2	188,358	280,326	288,370
B3	188,358	280,326	288,370
B4	622,008	713,976	723,520
B5	83,430	99,270	101,714

Table 2: Number of trainable parameters for each convolutional neural network architecture during transfer learning and fine tuning. Total values include trainable and non trainable parameters (convolutions' biases).

B.4. Effect of data sampling in model training

In Figure 7, we show a sanity experiment performed with our sampling strategy. We trained the same network architecture with three different approaches: (1) On each iteration the data generator crops a random patch from an image and introduces it into the network (no-sampling (no DA)); (2) same as before but the patch is randomly transformed for data augmentation (no-sampling (DA)); (3) first the original image is randomly transformed and then, a patch is cropped using a sampling probability distribution function over the pixels in the image. The loss function when there is no sampling probability distribution function (no-sampling) is lower than when it is applied (sampling run 1 and sampling run 2). However, the accuracy is less than 0.5 for the foreground in those cases, indicating that the network classifies most of the pixels as background. During the training, CNN average the loss function values for all the pixels in the input patch. Because the foreground-background ratio is quite low, classifying all the pixels as background leads to a fickle local minimum.

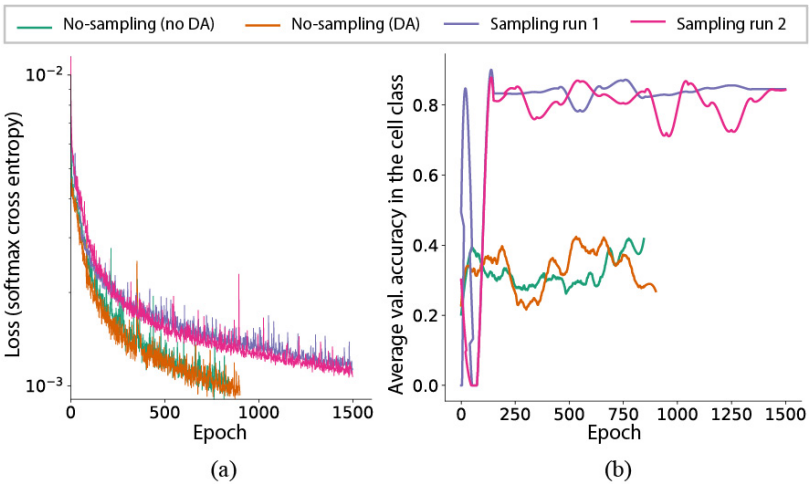

Figure 7: Effect of training data sampling strategy. MobileUNet network (Seif, 2018; Howard et al., 2017) trained with the same parameters and changing the training data sampling strategy. (a) Softmax cross-entropy loss function for each training epoch. (b) Averaged accuracy of the foreground (pixels classified as cells) for the validation dataset on each 50 epochs during the training. DA: data-augmentation.

When using a sampling probability distribution function, such optimal modes are avoided. Moreover, it takes much longer until the loss function converges to a value similar to the one for which all the pixels are classified as background. The result, thus, is that the network learns to classify those poorly represented pixels with very little programming and mathematical effort.

B.5. Stability of the learning process

See Figures 8 and 9.

B.6. Learning process of different model configurations

See Figure 10.

Appendix C. Protrusion tip quantification

See Figures 12 and 13.

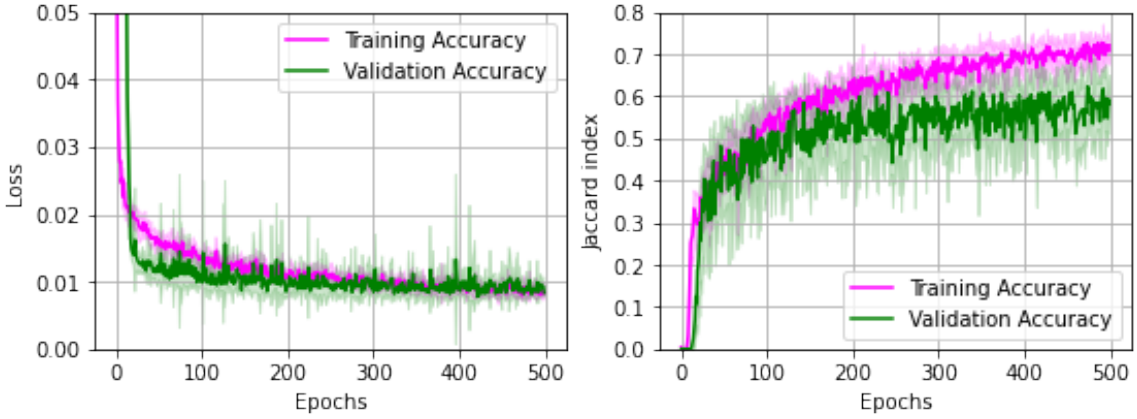

Figure 8: Replication of the learning process for the architecture B5 with exactly the same decoder initialization but randomly updating the training patches.

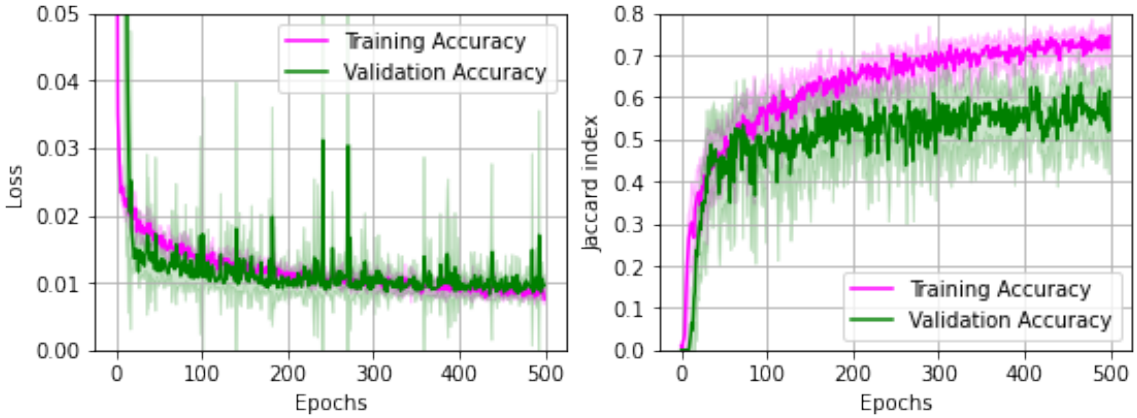

Figure 9: Replication of the learning process for the architecture B5 with a random decoder initialization but fixed training data patches.

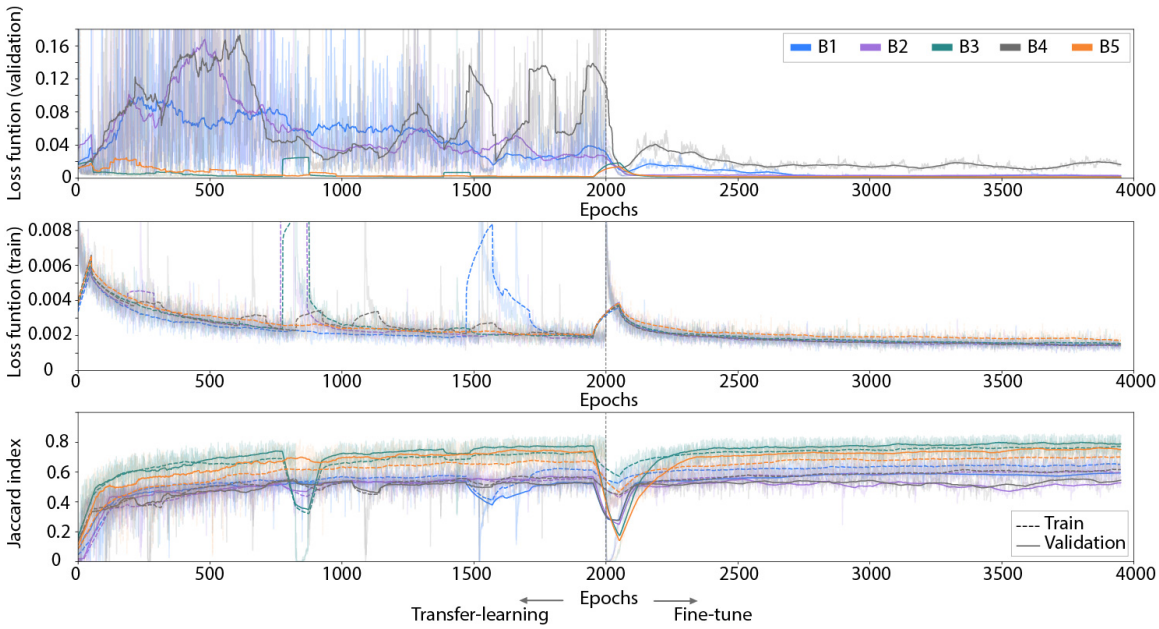

Figure 10: Learning process of the evaluated neural network architectures: (a) Architectures trained to segment cells (B1, B2, B3, B4, and B5). During the first $2K$ epochs, the pre-trained encoder was frozen and the decoder is trained using a learning rate of 0.005. During the remaining $2K$ epochs, the gradients are back-propagated to all the weights in the model using a learning rate of 0.0001. Training curves are smoothed to improve visualization.

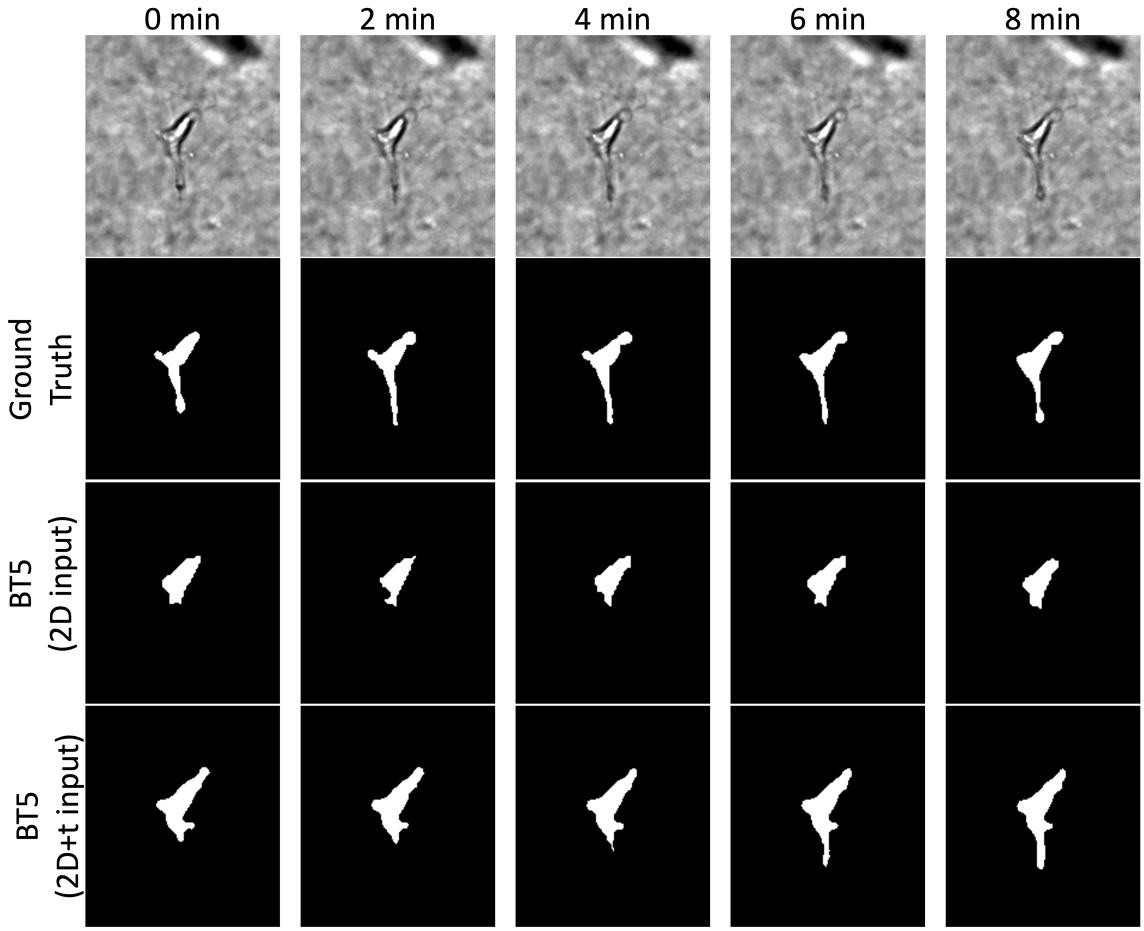

Figure 11: Results for the same CNN architecture (B5) having only the image at time t as input (B5 (2D input)) or a time series consisting of 5 frames previous to the frame at time t (B5 (2D + t input)). The images at $t = 0$ do not have previous frames so it is not until $t = 6$ min that the model can determine that the uncertain part is indeed a protrusion of the cell and needs to be segmented.

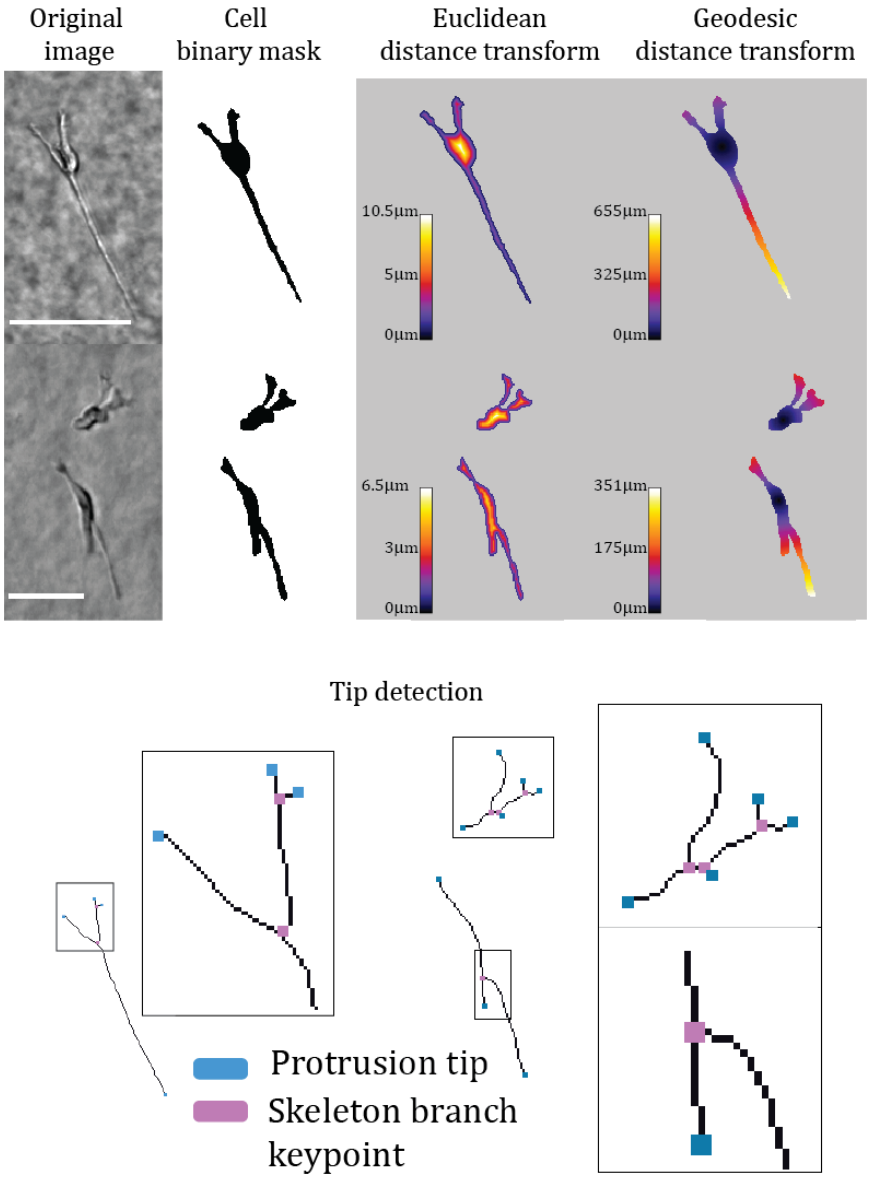

Figure 12: (Top) Image processing schema based on the cellular shape for the detection of protrusion tips. The local maximum of the Euclidean distance transform provides the cell body center. This reference point is employed for the calculation of the Geodesic distance transform. (Bottom) Sample images for the detection of protrusion tips (the extreme of the skeleton) and the skeleton branch keypoints. The images on the right correspond to the zoomed crops of the regions enclosed in the boxes on the left. Scale bars of 100 μm and 50 μm for the first and the second row, respectively.

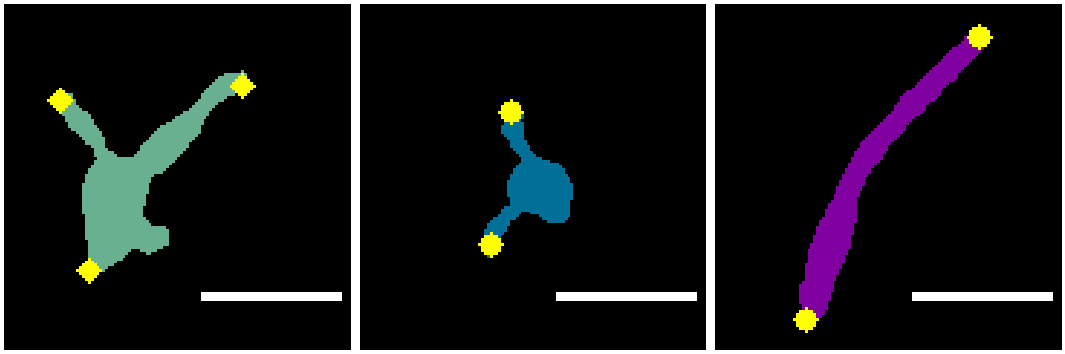

Figure 13: Result of the protrusion tip detection for different cell morphologies. The tips are labelled in yellow and the cell body in green, blue and violet, respectively. Scale bars of 40 μm.