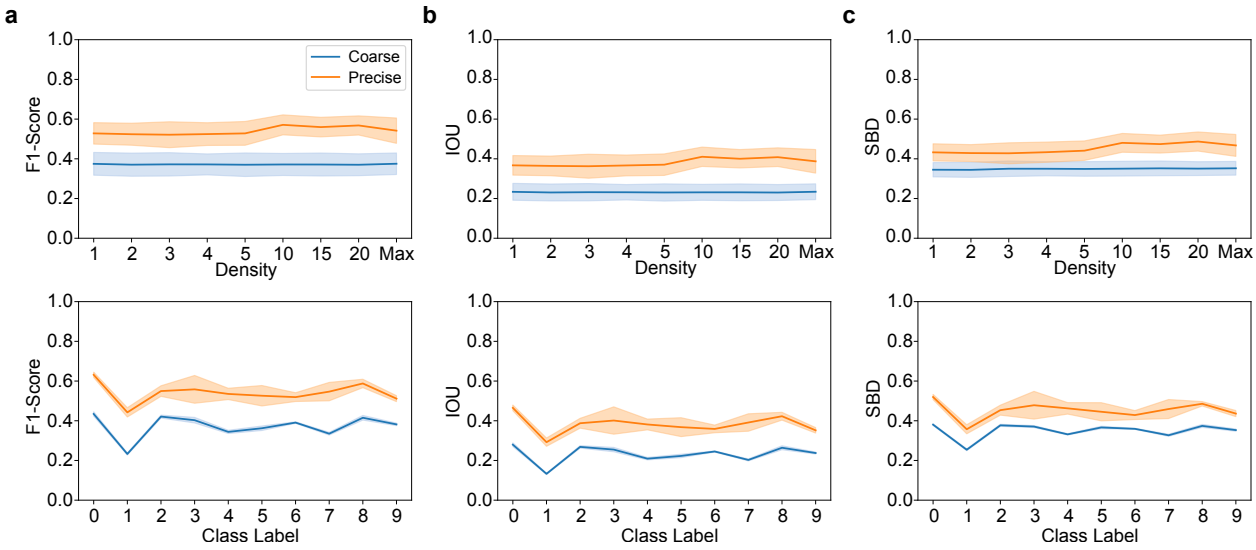
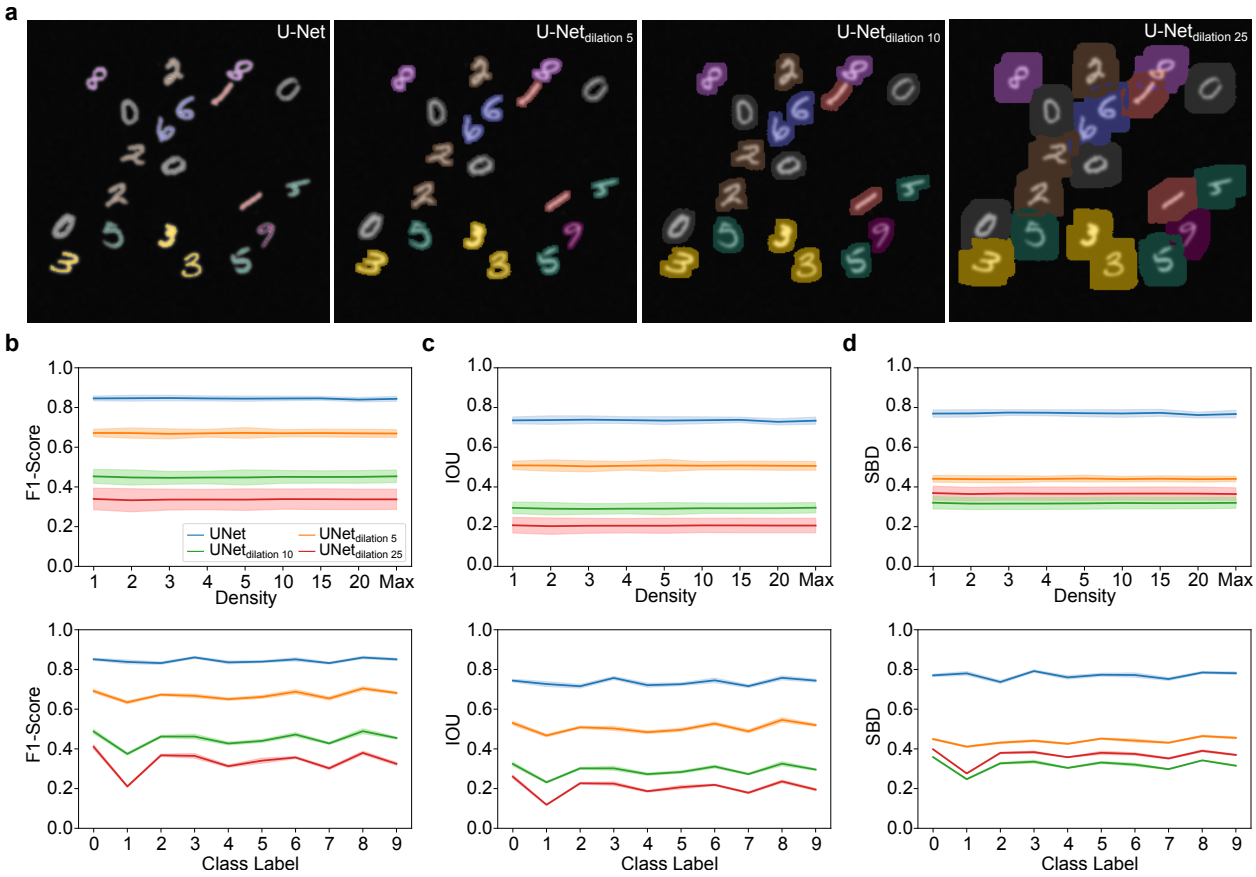
MICRA-Net: MICRoscopy Analysis Neural Network to solve detection, classification, and segmentation from a single simple auxiliary task

Supplementary Figures and Tables



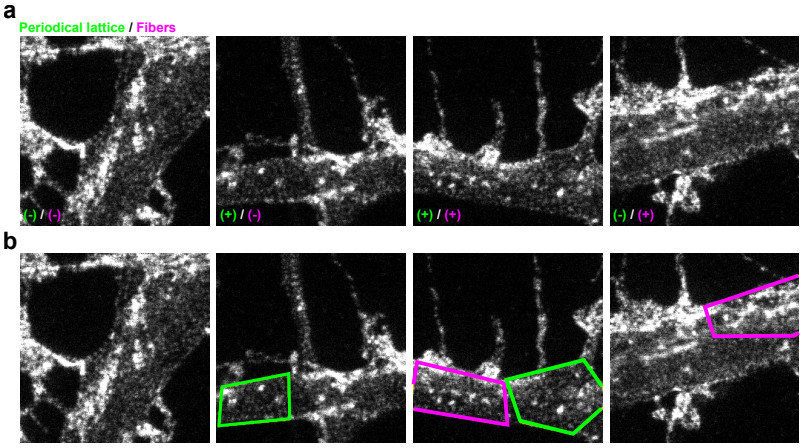
Supplementary Figure 1: Comparison of the coarse and precise segmentation on the modified MNIST dataset for three different metrics a) F1-Score, b) intersection over union (IOU), and c) symmetric boundary dice (SBD). Top row presents the metric as a function of the numbers of digits in the field of view (Density) while bottom row shows the metrics performance in a per class manner. Solid lines and pale regions represents the mean and standard deviation respectively.



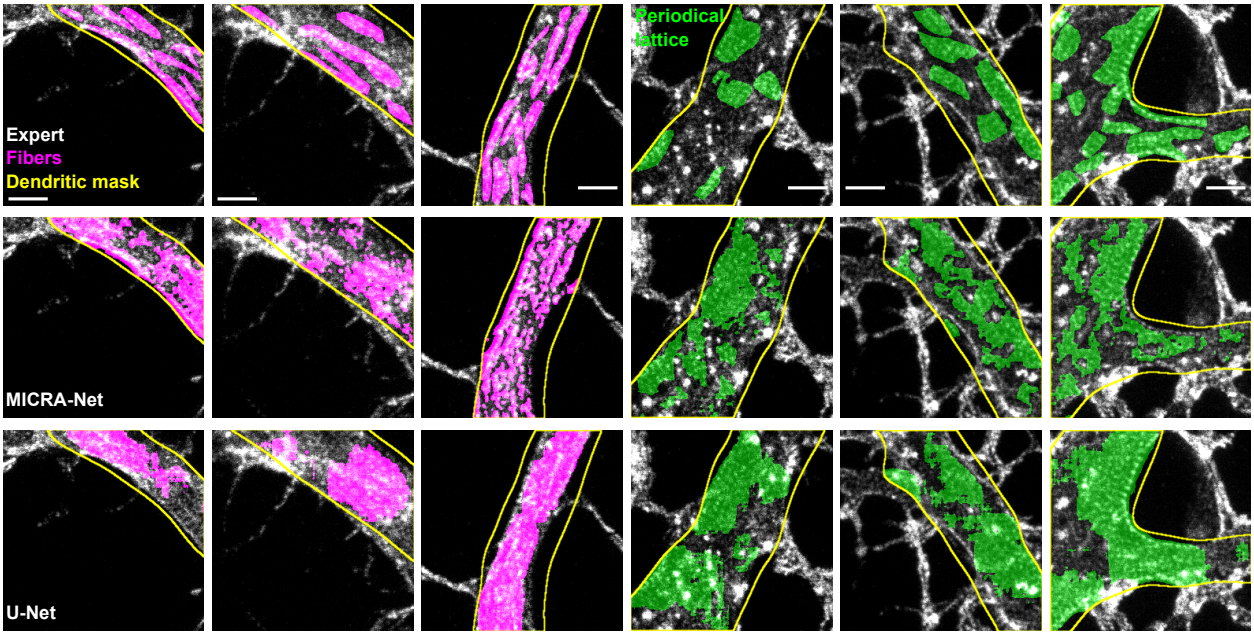
Supplementary Figure 2: U-Net performance trained in a fully- and weakly-supervised fashion on the modified MNIST dataset. a) Segmentation examples taken from the testing dataset with an increasing dilation of the MNIST binary digits to simulate weak supervision. The dilation are computed using a square structuring element of $\{5, 10, 25\}$ pixel. Metric performance b) F1-score, c) intersection over union (IOU), and d) symmetric boundary dice (SBD) are assessed and compared for the different training scheme. Top and bottom rows compare the per density and per class performance respectively. Solid lines and pale regions represents the mean and standard deviation respectively.

F1-score	Coarse	Precise	U-Net	U-Net5	U-Net10	U-Net25
Coarse	-	1.6616×10^{-6}	1.8475×10^{-15}	1.0448×10^{-11}	0.0016	0.1724
Precise	1.6616×10^{-6}	-	2.0254×10^{-12}	3.6243×10^{-6}	1.4390×10^{-4}	7.3312×10^{-8}
U-Net	1.8475×10^{-15}	2.0254×10^{-12}	-	8.1778×10^{-15}	5.1211×10^{-18}	1.5283×10^{-16}
U-Net5	1.0448×10^{-11}	3.6243×10^{-6}	8.1778×10^{-15}	-	9.4503×10^{-16}	5.2542×10^{-13}
U-Net10	0.0016	1.4390×10^{-4}	5.1211×10^{-18}	9.4503×10^{-16}	-	2.2030×10^{-5}
U-Net25	0.1724	7.3312×10^{-8}	1.5283×10^{-16}	5.2542×10^{-13}	2.2030×10^{-5}	-
IOU	Coarse	Precise	U-Net	U-Net5	U-Net10	U-Net25
Coarse	-	5.2650×10^{-7}	6.3229×10^{-18}	7.1191×10^{-13}	0.0013	0.1525
Precise	5.2650×10^{-7}	-	8.7200×10^{-14}	4.5055×10^{-6}	4.4700×10^{-5}	3.0355×10^{-8}
U-Net	6.3229×10^{-18}	8.7200×10^{-14}	-	2.4485×10^{-15}	1.3616×10^{-19}	5.1516×10^{-19}
U-Net5	7.1191×10^{-13}	4.5055×10^{-6}	2.4485×10^{-15}	-	3.6827×10^{-13}	3.7153×10^{-14}
U-Net10	0.0013	4.4700×10^{-5}	1.3616×10^{-19}	3.6827×10^{-13}	-	1.3703×10^{-5}
U-Net25	0.1525	3.0355×10^{-8}	5.1516×10^{-19}	3.7153×10^{-14}	1.3703×10^{-5}	-
SBD	Coarse	Precise	U-Net	U-Net5	U-Net10	U-Net25
Coarse	-	1.1627×10^{-5}	2.7395×10^{-17}	1.5261×10^{-6}	0.0459	0.3958
Precise	1.1627×10^{-5}	-	5.7291×10^{-13}	0.1032	1.7367×10^{-7}	3.5163×10^{-5}
U-Net	2.7395×10^{-17}	5.7291×10^{-13}	-	8.3720×10^{-20}	3.2693×10^{-19}	1.2818×10^{-17}
U-Net5	1.5261×10^{-6}	0.1032	8.3720×10^{-20}	-	9.7111×10^{-10}	4.7895×10^{-6}
U-Net10	0.0459	1.7367×10^{-7}	3.2693×10^{-19}	9.7111×10^{-10}	-	4.5237×10^{-3}
U-Net25	0.3958	3.5163×10^{-5}	1.2818×10^{-17}	4.7895×10^{-6}	4.5237×10^{-3}	-

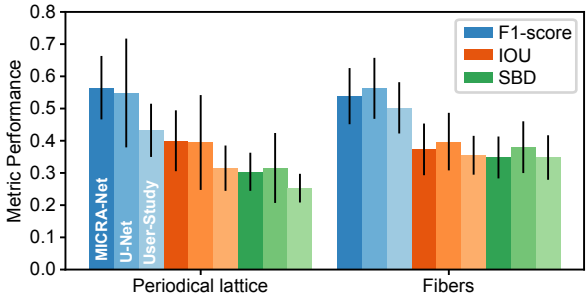
Supplementary Table 1: Comparison of the performance metrics (F1-score, IOU, and SBD) for MICRA-Net (coarse and precise) and U-Net (weakly- and fully-supervised) segmentation on the modified MNIST dataset. The p-values are obtained by a posthoc t-test following a one way ANOVA ($p_{\mathrm{F1-score}} = 9.9543 \times 10^{-34}$, $p_{\mathrm{IOU}} = 8.1207 \times 10^{-39}$ and $p_{\mathrm{SBD}} = 3.1213 \times 10^{-36}$). Color code: increase (green), decrease (red), and no significant changes (black) of the metric scores.



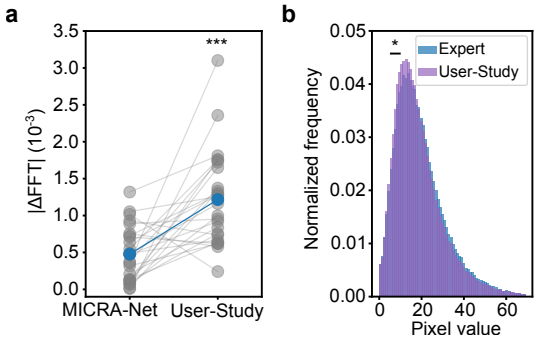
Supplementary Figure 3: Representative examples of 256×256 pixel crops sampled from the training set of the F-actin dataset. We present in a) positive (+) and negative (-) crops, and in b) the associated polygonal bounding box annotations. The F-actin periodical lattice is in green, while longitudinal fibers are depicted in magenta. Each crop is 3.84×3.84 µm.



Supplementary Figure 4: Representative images of F-actin semantic segmentation on dendrites. From top to bottom : 1) precise expert annotations, 2) MICRA-Net predictions, and 3) weakly-supervised U-Net predictions. As shown, MICRA-Net can generate a semantic segmentation that is comparable to the expert annotations for both structures. Scale bars $1\mu m$.



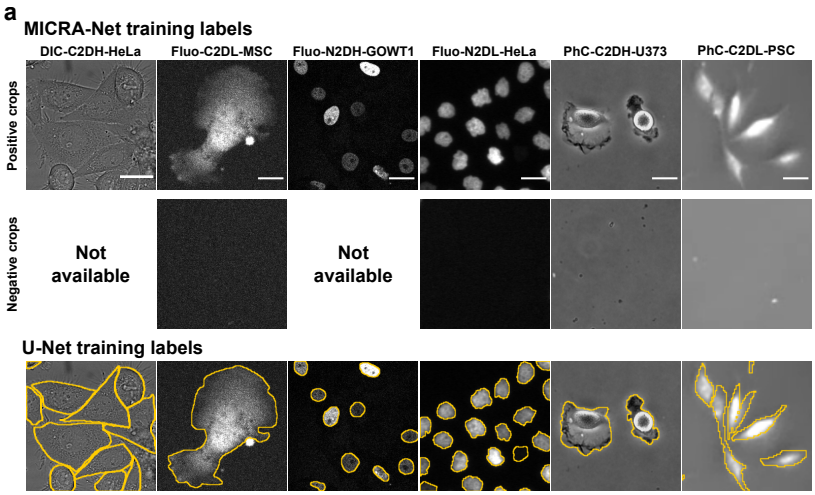
Supplementary Figure 5: F1-Score (blue), IOU (orange), and SBD (green) metrics for MICRA-Net (dark) weakly-supervised U-Net (medium), and user-study (light) segmentation masks calculated using a precisely annotated testing set of 50 images as a ground truth. No statistical differences is measured for F1-score and IOU (one-way ANOVA, $p_{\text{F1-score}} = 0.1801$ and $p_{\text{IOU}} = 0.4820$) for both structures, while a significant increase is noted in SBD for fibers over periodical lattice for all methods (one-way ANOVA : $p_{\text{SBD}} = 0.0023$ and post-hoc t-test : $p_{\text{SBD, MICRA-Net, pl-fibers}} = 0.0169$, $p_{\text{SBD, U-Net, pl-fibers}} = 0.0234$, $p_{\text{SBD, User-Study, pl-fibers}} = 0.0271$). Error bars (black lines) correspond to standard deviation.



Supplementary Figure 6: Comparison between the user-study and MICRA-Net on the segmentation of F-actin nanostructures using user defined metrics. a) A significant decrease of the FTT metric is observed for the MICRA-Net compared to the user-study implying in a smaller different with precise expert annotations, and therefore a more accurate segmentation (t-test, $p = 7.9298 \times 10^{-6}$). b) The pixel intensity distribution metric for the user-study was calculated as the average histogram of annotated pixels for the 6 experts. A statistical analysis revealed a significant difference in the number of low-intensity ($[5, 9]$) foreground pixels values for the user-study when compared to the expert annotations, while no difference is observed for MICRA-Net (Figure 3 and Supplementary Tab. 2 for p-values).

Pixel value	0	1	2	3	4	5	6	7	8	9
MICRA-Net	0.1304	0.1835	0.1671	0.12712	0.1720	0.1608	0.0998	0.0633	0.0910	0.0676
U-Net	0.2526	0.9271	0.2924	0.0989	0.0368	0.0169	0.0095	0.0044	0.0046	0.0027
User-Study	0.3484	0.3049	0.1751	0.1091	0.0667	0.0421	0.0274	0.0147	0.0204	0.0177

Supplementary Table 2: Comparison of the expert annotations distribution of intensity and the predicted segmentation masks for the fiber metric. Only pixel values that are considered low-intensity are shown (see Methods). The p-values are obtained from a t-test comparing the distribution of normalized pixel counts in a given intensity bin. Color code: increase (red), and no significant changes (black) of the fiber metric. An increase corresponds to a significant difference with the ground truth (precise expert annotations).



b

Segmentation (IOU)

	U-Net	MICRA-Net
C2DH-HeLa	0.75 ± 0.07	0.10 ± 0.02
C2DL-MSC	0.47 ± 0.11	0.34 ± 0.05
N2DH-GOWT1	0.59 ± 0.13	0.29 ± 0.03
N2DL-HeLa	0.84 ± 0.01	0.60 ± 0.05
C2DH-U373	0.88 ± 0.03	0.64 ± 0.01
C2DL-PSC	0.65 ± 0.01	0.53 ± 0.03

Supplementary Figure 7: Evaluation of the segmentation performance of MICRA-Net compared to fully-supervised, pre-trained U-Net using the same resampling method of images as Falk et al. [1] on the Cell Tracking Challenge (CTC). While MICRA-Net obtained a high classification accuracy ($(95.8 \pm 0.4)\,\%$) on the testing set, it minimally requires examples of negative crops to be able to extract enough context to differentiate between the cells and background. This is demonstrated by the poor segmentation performance on DIC-C2DH-HeLa and Fluo-N2DH-GOWT1 cell lines when no negative crops are extracted.

Cell line	U-Net	MICRA-Net
DIC-C2DH-HeLa	0.38	1.00
Fluo-C2DL-MSC	0.79	1.28
Fluo-N2DH-GOWT1	0.48	1.00
Fluo-N2DL-HeLa	1.29	2.15
PhC-C2DH-U373	1.30	1.30
PhC-C2DH-PSC	3.20	3.20

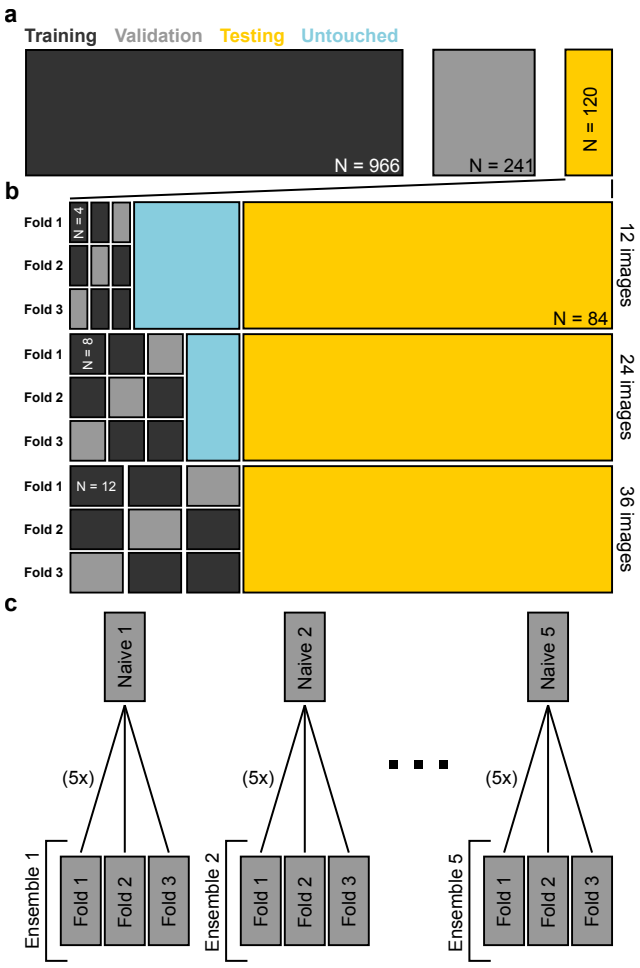
Supplementary Table 3: Scale factors used to resize the CTC cell lines images for MICRA-Net and fully-supervised pre-trained U-Net training.

Dataset	p-value
DIC-C2DH-HeLa	0.8544
Fluo-C2DL-MSC	0.9402
Fluo-N2DH-GOWT1	0.4031
Fluo-N2DL-HeLa	0.0062
PhC-C2DH-U373	0.0306
PhC-C2DL-PSC	0.0089

Supplementary Table 4: MICRA-Net and fully-supervised U-Net segmentation performance on the CTC dataset. For half of the cell lines, when comparing U-Net and MICRA-Net, a significantly increased U-Net segmentation performance is measured (red), while for the other half, the two networks obtain similar performances. The p-values were calculated using resampling (see Methods) from 5 different network instantiations.

Dataset	p-value
DIC-C2DH-HeLa	0.8116
Fluo-C2DL-MSC	0.0024
Fluo-N2DH-GOWT1	0.0179
Fluo-N2DL-HeLa	0.0075
PhC-C2DH-U373	0.0211
PhC-C2DL-PSC	0.0064

Supplementary Table 5: MICRA-Net and fully-supervised U-Net detection performance on the CTC dataset. Compared with U-Net detection performance, MICRA-Net shows an improvement for 4 cell lines (green), similar performance for 1 cell line (black), and an decrease for 1 cell line (red). The p-values were calculated using resampling (see Methods) from 5 different network instantiations.



Supplementary Figure 8: Schematic of the training and fine-tuning procedure for MICRA-Net on the P. vivax dataset. a) Data preparation: 80/20 split of the provided training set for training and validation respectively, keeping the testing set as is. b) Fine-tuning of MICRA-Net: uniform sample of $\{12, 24, 36\}$ images from the testing set. A 3-fold scheme: training on two folds and validating on a separate fold, enabling early stopping. The average prediction of the 3 fine-tuned models (termed ensemble) was used for testing. All methods were tested on the same testing set of 84 images. c) Training: 5 different models were trained on the original dataset (Naive). For fine-tuning, the 3-fold scheme was repeated 5 times, one time for each of the 5 Naive models as starting points, generating a total of 25 ensemble models.

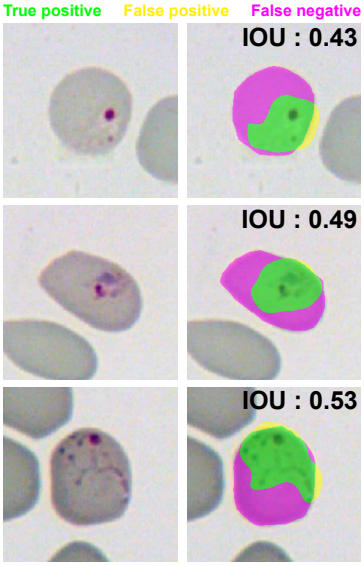
Condition (Number of images)	Accuracy
Naive (-)	$(0.8 \pm 0.1)\,\%$
Threshold (12)	$(0.88 \pm 0.02)\,\%$
Threshold (24)	$(0.87 \pm 0.02)\,\%$
Threshold (36)	$(0.87 \pm 0.02)\,\%$
Linear (12)	$(0.893 \pm 0.008)\,\%$
Linear (24)	$(0.888 \pm 0.007)\,\%$
Linear (36)	$(0.890 \pm 0.005)\,\%$
Linear + 4 (12)	$(0.909 \pm 0.005)\,\%$
Linear + 4 (24)	$(0.904 \pm 0.003)\,\%$
Linear + 4 (36)	$(0.905 \pm 0.003)\,\%$
Linear + 3, 4 (12)	$(0.911 \pm 0.003)\,\%$
Linear + 3, 4 (24)	$(0.904 \pm 0.004)\,\%$
Linear + 3, 4 (36)	$(0.906 \pm 0.003)\,\%$
All (12)	$(0.908 \pm 0.005)\,\%$
All (24)	$(0.906 \pm 0.003)\,\%$
All (36)	$(0.906 \pm 0.003)\,\%$

Supplementary Table 6: Classification accuracy of the naive and fine-tuned models on their respective testing set on the P. vivax dataset. The number of images used to adjust the thresholds or fine-tuning is in parentheses. The accuracy is reported as the mean ± standard deviation. For the network configuration Naive and Threshold $\{12, 24, 36\}$ (see Methods) it is calculated from 5 network instantiations. For the fine-tuned models, the scores from the ensembles that were fine-tuned from a single Naive model were averaged generating 5 classification scores that are used for calculation.

F1-score	Linear (12)	Linear (24)	Linear (36)	Linear + 4 (12)	Linear + 4 (24)	Linear + 4 (36)	Linear + 3, 4 (12)	Linear + 3, 4 (24)	Linear + 3, 4 (36)	All (12)	All (24)	All (36)	Naive	Threshold (12)	Threshold (24)	Threshold (36)
Linear (12)	-	0.7972	0.2409	0.0043	0.0055	0.0062	0.0016	0.0020	0.0020	0.0021	0.0015	0.0043	0.0241	0.1812	0.1539	0.2185
Linear (24)	0.7972	-	0.3826	0.0086	0.0091	0.0096	0.0071	0.0032	0.0060	0.0068	0.0049	0.0093	0.0099	0.1183	0.1009	0.1578
Linear (36)	0.2409	0.3826	-	0.0040	0.0057	0.0059	0.0030	0.0011	0.0028	0.0024	0.0018	0.0026	0.0087	0.0166	0.0197	0.0522
Linear + 4 (12)	0.0043	0.0086	0.0040	-	0.0070	0.0078	0.0038	0.0033	0.0047	0.0048	0.0021	0.0068	0.0085	0.0066	0.0025	0.0045
Linear + 4 (24)	0.0055	0.0091	0.0057	0.0070	-	0.0312	0.9026	0.0725	0.0023	0.7719	0.0267	0.0036	0.0084	0.0083	0.0036	0.0051
Linear + 4 (36)	0.0062	0.0096	0.0059	0.0078	0.0312	-	0.0414	0.1689	0.0273	0.0549	0.7127	0.0137	0.0085	0.0065	0.0043	0.0049
Linear + 3, 4 (12)	0.0016	0.0071	0.0030	0.0038	0.9026	0.0414	-	0.0595	0.0042	0.7141	0.0124	0.0088	0.0043	0.0036	0.0010	0.0024
Linear + 3, 4 (24)	0.0020	0.0032	0.0011	0.0033	0.0725	0.1689	0.0595	-	0.0061	0.1468	0.1310	0.0105	0.0029	0.0025	0.0006	0.0015
Linear + 3, 4 (36)	0.0020	0.0060	0.0028	0.0047	0.0023	0.0273	0.0042	0.0061	-	0.0043	0.0043	0.5320	0.0049	0.0033	0.0008	0.0015
All (12)	0.0021	0.0068	0.0024	0.0048	0.7719	0.0549	0.7141	0.1468	0.0043	-	0.0537	0.0077	0.0064	0.0065	0.0022	0.0028
All (24)	0.0015	0.0049	0.0018	0.0021	0.0267	0.7127	0.0124	0.1310	0.0043	0.0537	-	0.0084	0.0038	0.0030	0.0006	0.0011
All (36)	0.0043	0.0093	0.0026	0.0068	0.0036	0.0137	0.0088	0.0105	0.5320	0.0077	0.0084	-	0.0066	0.0062	0.0029	0.0042
Naive	0.0241	0.0099	0.0087	0.0085	0.0084	0.0085	0.0043	0.0029	0.0049	0.0064	0.0038	0.0066	-	0.0933	0.1115	0.0954
Threshold (12)	0.1812	0.1183	0.0166	0.0066	0.0083	0.0065	0.0036	0.0025	0.0033	0.0065	0.0030	0.0062	0.0933	-	0.9359	0.8587
Threshold (24)	0.1539	0.1009	0.0197	0.0025	0.0036	0.0043	0.0010	0.0006	0.0008	0.0022	0.0006	0.0029	0.1115	0.9359	-	0.9287
Threshold (36)	0.2185	0.1578	0.0522	0.0045	0.0051	0.0049	0.0024	0.0015	0.0015	0.0028	0.0011	0.0042	0.0954	0.8587	0.9287	-

Supplementary Table 7: F1-score detection metric for the fine-tuned networks on the P. vivax dataset. The F-statistic from all groups was bootstrapped resulting in a p-value of 0. A post-hoc resampling statistical test was performed to compare the distributions of each groups in a one-to-one manner (see Methods). Color code: increase (green), decrease (red), and no significant changes (black) in the F1-score.

a



b

Segmentation (F1-score)

	12 images	24 images	36 images
Linear	0.66 ± 0.06	0.67 ± 0.06	0.67 ± 0.06
Linear + 4	0.61 ± 0.02	0.62 ± 0.02	0.626 ± 0.007
Linear + 3, 4	0.59 ± 0.01	0.61 ± 0.02	0.61 ± 0.02
All	0.59 ± 0.02	0.60 ± 0.02	0.62 ± 0.02
Naive	0.55 ± 0.07		

Segmentation (IOU)

	12 images	24 images	36 images
Linear	0.52 ± 0.06	0.53 ± 0.06	0.53 ± 0.06
Linear + 4	0.47 ± 0.02	0.48 ± 0.02	0.48 ± 0.007
Linear + 3, 4	0.44 ± 0.02	0.46 ± 0.02	0.47 ± 0.02
All	0.44 ± 0.02	0.46 ± 0.02	0.48 ± 0.03
Naive	0.43 ± 0.06		

Segmentation (SBD)

	12 images	24 images	36 images
Linear	0.299 ± 0.006	0.301 ± 0.007	0.285 ± 0.002
Linear + 4	0.285 ± 0.003	0.288 ± 0.003	0.287 ± 0.006
Linear + 3, 4	0.278 ± 0.007	0.282 ± 0.008	0.283 ± 0.008
All	0.277 ± 0.009	0.281 ± 0.009	0.286 ± 0.01
Naive	0.29 ± 0.01		

Supplementary Figure 9: Evaluation of the segmentation performance of MICRA-Net on the P. vivax dataset. a) Example of segmentation with three different IOU. The IOU were chosen according to the average of IOU in b). Color code: True positive (green), false positive (yellow), and false negatives (magenta). b) The 3 common segmentation scores (F1-score, IOU, and SBD) were compared between the fine-tuned and Naive models. The statistical analysis is presented in Supplementary Tab. 8. The presented scores are mean ± 95% confidence interval bootstrapped from the trained and fine-tuned models.

F1-score	Naive	Linear (12)	Linear (24)	Linear (36)	Linear + 4 (12)	Linear + 4 (24)	Linear + 4 (36)	Linear + 3, 4 (12)	Linear + 3, 4 (24)	Linear + 3, 4 (36)	All (12)	All (24)	All (36)
Naive	-	0.0095	0.0042	0.0039	0.0192	0.0084	0.0091	0.1674	0.0269	0.0428	0.1895	0.0712	0.0213
Linear (12)	0.0095	-	0.7434	0.7344	0.0975	0.1521	0.1480	0.0085	0.0520	0.0512	0.0172	0.0313	0.1055
Linear (24)	0.0042	0.7434	-	0.9331	0.0364	0.0721	0.0791	0.0047	0.0128	0.0262	0.0049	0.0131	0.0534
Linear (36)	0.0039	0.7344	0.9331	-	0.0457	0.1007	0.0968	0.0035	0.0313	0.0338	0.0115	0.0190	0.0715
Linear + 4 (12)	0.0192	0.0975	0.0364	0.0457	-	0.4275	0.2661	0.0358	0.6065	0.6760	0.0399	0.2724	0.8483
Linear + 4 (24)	0.0084	0.1521	0.0721	0.1007	0.4275	-	0.8856	0.0065	0.1707	0.2480	0.0254	0.0875	0.5632
Linear + 4 (36)	0.0091	0.1480	0.0791	0.0968	0.2661	0.8856	-	0.0077	0.0411	0.1194	0.0073	0.0312	0.3635
Linear + 3, 4 (12)	0.1674	0.0085	0.0047	0.0035	0.0358	0.0065	0.0077	-	0.0343	0.0752	0.9831	0.1463	0.0272
Linear + 3, 4 (24)	0.0269	0.0520	0.0128	0.0313	0.6065	0.1707	0.0411	0.0343	-	0.9686	0.0797	0.4145	0.5521
Linear + 3, 4 (36)	0.0428	0.0512	0.0262	0.0338	0.6760	0.2480	0.1194	0.0752	0.9686	-	0.0944	0.4940	0.5785
All (12)	0.1895	0.0172	0.0049	0.0115	0.0399	0.0254	0.0073	0.9831	0.0797	0.0944	-	0.2135	0.0553
All (24)	0.0712	0.0313	0.0131	0.0190	0.2724	0.0875	0.0312	0.1463	0.4145	0.4940	0.2135	-	0.2313
All (36)	0.0213	0.1055	0.0534	0.0715	0.8483	0.5632	0.3635	0.0272	0.5521	0.5785	0.0553	0.2313	-

IOU	Naive	Linear (12)	Linear (24)	Linear (36)	Linear + 4 (12)	Linear + 4 (24)	Linear + 4 (36)	Linear + 3, 4 (12)	Linear + 3, 4 (24)	Linear + 3, 4 (36)	All (12)	All (24)	All (36)
Naive	-	0.0107	0.0159	0.0147	0.0973	0.0540	0.0145	0.6296	0.1416	0.1146	0.5647	0.1955	0.0620
Linear (12)	0.0107	-	0.7089	0.7211	0.0614	0.1067	0.1344	0.0018	0.0322	0.0440	0.0115	0.0355	0.1242
Linear (24)	0.0159	0.7089	-	0.9160	0.0321	0.0608	0.0676	0.0016	0.0111	0.0340	0.0037	0.0085	0.0561
Linear (36)	0.0147	0.7211	0.9160	-	0.0431	0.0578	0.0632	0.0020	0.0173	0.0315	0.0109	0.0163	0.0653
Linear + 4 (12)	0.0973	0.0614	0.0321	0.0431	-	0.4465	0.2202	0.0393	0.6931	0.9140	0.0651	0.4117	0.5429
Linear + 4 (24)	0.0540	0.1067	0.0608	0.0578	0.4465	-	0.7080	0.0111	0.2176	0.3453	0.0350	0.1433	0.9274
Linear + 4 (36)	0.0145	0.1344	0.0676	0.0632	0.2202	0.7080	-	0.0059	0.0613	0.1558	0.0068	0.0339	0.7059
Linear + 3, 4 (12)	0.6296	0.0018	0.0016	0.0020	0.0393	0.0111	0.0059	-	0.0346	0.0421	0.9614	0.1089	0.0120
Linear + 3, 4 (24)	0.1416	0.0322	0.0111	0.0173	0.6931	0.2176	0.0613	0.0346	-	0.8171	0.0928	0.6394	0.2773
Linear + 3, 4 (36)	0.1146	0.0440	0.0340	0.0315	0.9140	0.3453	0.1558	0.0421	0.8171	-	0.0834	0.5150	0.4615
All (12)	0.5647	0.0115	0.0037	0.0109	0.0651	0.0350	0.0068	0.9614	0.0928	0.0834	-	0.2094	0.0550
All (24)	0.1955	0.0355	0.0085	0.0163	0.4117	0.1433	0.0339	0.1089	0.6394	0.5150	0.2094	-	0.1811
All (36)	0.0620	0.1242	0.0561	0.0653	0.5429	0.9274	0.7059	0.0120	0.2773	0.4615	0.0550	0.1811	-

SBD	Naive	Linear (12)	Linear (24)	Linear (36)	Linear + 4 (12)	Linear + 4 (24)	Linear + 4 (36)	Linear + 3, 4 (12)	Linear + 3, 4 (24)	Linear + 3, 4 (36)	All (12)	All (24)	All (36)
Naive	-	0.0665	0.0446	0.0466	0.3973	0.7636	0.6791	0.0501	0.2301	0.3394	0.0631	0.1584	0.6486
Linear (12)	0.0665	-	0.5449	0.5658	0.0076	0.0070	0.0152	0.0092	0.0090	0.0180	0.0080	0.0087	0.0405
Linear (24)	0.0446	0.5449	-	0.8918	0.0074	0.0086	0.0079	0.0087	0.0085	0.0086	0.0085	0.0081	0.0302
Linear (36)	0.0466	0.5658	0.8918	-	0.0088	0.0084	0.0093	0.0055	0.0094	0.0083	0.0084	0.0083	0.0332
Linear + 4 (12)	0.3973	0.0076	0.0074	0.0088	-	0.0964	0.4840	0.0172	0.4120	0.7118	0.0576	0.2452	0.8397
Linear + 4 (24)	0.7636	0.0070	0.0086	0.0084	0.0964	-	0.7498	0.0074	0.0470	0.2947	0.0202	0.1216	0.7521
Linear + 4 (36)	0.6791	0.0152	0.0079	0.0093	0.4840	0.7498	-	0.0302	0.1968	0.4225	0.0583	0.1747	0.9009
Linear + 3, 4 (12)	0.0501	0.0092	0.0087	0.0055	0.0172	0.0074	0.0302	-	0.3277	0.1936	0.7907	0.5190	0.1573
Linear + 3, 4 (24)	0.2301	0.0090	0.0085	0.0094	0.4120	0.0470	0.1968	0.3277	-	0.7440	0.2827	0.7772	0.4555
Linear + 3, 4 (36)	0.3394	0.0180	0.0086	0.0083	0.7118	0.2947	0.4225	0.1936	0.7440	-	0.1665	0.5405	0.6670
All (12)	0.0631	0.0080	0.0085	0.0084	0.0576	0.0202	0.0583	0.7907	0.2827	0.1665	-	0.4252	0.1450
All (24)	0.1584	0.0087	0.0081	0.0083	0.2452	0.1216	0.1747	0.5190	0.7772	0.5405	0.4252	-	0.3625
All (36)	0.6486	0.0405	0.0302	0.0332	0.8397	0.7521	0.9009	0.1573	0.4555	0.6670	0.1450	0.3625	-

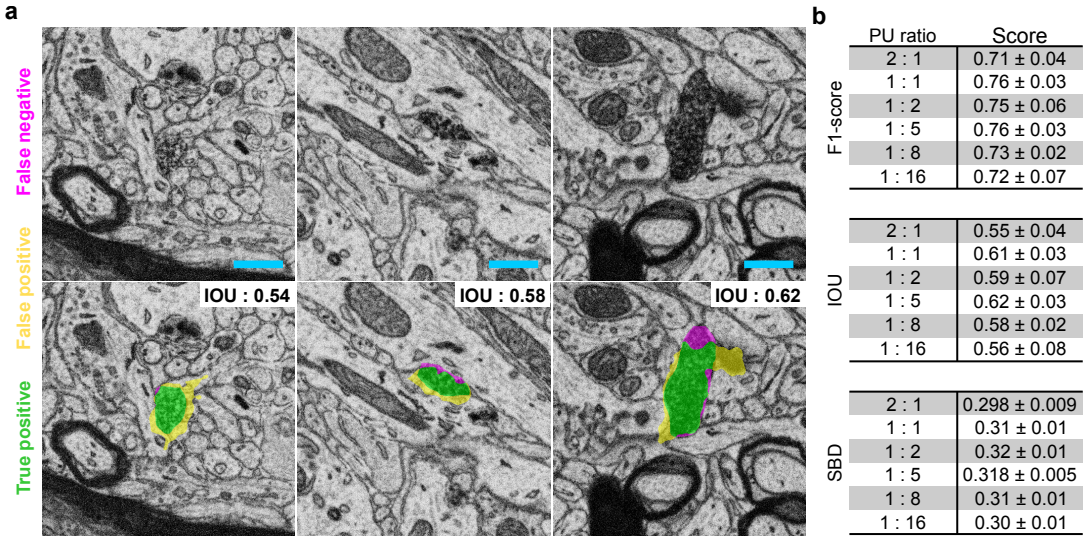
Supplementary Table 8: Statistical analysis on the comparison of the F1-score, IOU, and SBD segmentation metric for the fine-tuned networks for the P. vivax dataset. The main results are reported in Figure 9. The F-statistic from all groups was bootstrapped resulting in a p-value of 0. A post-hoc resampling statistical test was performed to compare the distributions of each groups in a one-to-one manner (see Methods). Color code: increase (green), decrease (red), and no significant changes (black) in the metrics scores.

Models	Accuracy
2 : 1	$(84 \pm 1)\,\%$
1 : 1	$(86.7 \pm 0.8)\,\%$
1 : 2	$(88 \pm 1)\,\%$
1 : 5	$(88.4 \pm 0.9)\,\%$
1 : 8	$(89 \pm 1)\,\%$
1 : 16	$(89.3 \pm 0.6)\,\%$

Supplementary Table 9: Classification accuracy of the trained models using different positive-unlabeled ratio on the scanning electron microscopy dataset. The accuracy is reported as the mean ± standard deviation calculated from 5 network models.

F1-score	2 : 1	1 : 1	1 : 2	1 : 5	1 : 8	1 : 16
2 : 1	-	0.0289	0.0050	0.0043	0.0055	0.0074
1 : 1	0.0289	-	0.1352	0.0231	0.1314	0.0474
1 : 2	0.0050	0.1352	-	0.2760	0.6855	0.4546
1 : 5	0.0043	0.0231	0.2760	-	0.3985	0.7141
1 : 8	0.0055	0.1314	0.6855	0.3985	-	0.6663
1 : 16	0.0074	0.0474	0.4546	0.7141	0.6663	-

Supplementary Table 10: Statistical analysis on the comparison of the F1-score detection metric of MICRA-Net for different positive-unlabeled ratios on the scanning electron microscopy dataset (Figure 6). The F-statistic from all groups was bootstrapped resulting in a p-value of 0. A post-hoc resampling statistical test was performed to compare the distributions of each groups in a one-to-one manner (see Methods). Color code: increase (green), decrease (red), and no significant changes (black) in the F1-score.



Supplementary Figure 10: Segmentation performance of MICRA-Net on the scanning electron microscopy dataset. a) Original images (top) with their corresponding segmentation (bottom). Scale bars are 1 µm. Color code : true positives (green), false positives (yellow), and false negatives (magenta). b) F1-score, IOU, and SBD evaluated on the precisely testing set. Reported is the mean ± 95% confidence interval from 5 network instantiations. A statistical difference was measured with resampling for SBD only ($p_{\text{SBD}} = 0.0059$, $p_{\text{F1-score}=0.1831}$, and $p_{\text{IOU}=0.1845}$). The post-hoc resampling statistical test for SBD is reported in Supplementary Tab. 11.

	2 : 1	1 : 1	1 : 2	1 : 5	1 : 8	1 : 16
2 : 1	-	0.0470	0.0210	0.0045	0.0358	0.6600
1 : 1	0.0470	-	0.2004	0.0541	0.7436	0.2544
1 : 2	0.0210	0.2004	-	0.7287	0.5321	0.0562
1 : 5	0.0045	0.0541	0.7287	-	0.3302	0.0203
1 : 8	0.0358	0.7436	0.5321	0.3302	-	0.1766
1 : 16	0.6600	0.2544	0.0562	0.0203	0.1766	-

Supplementary Table 11: Results from the statistical analysis comparison of the SBD segmentation metric the different positive-unlabeled ratio on the scanning electron microscopy dataset. The main results are reported in Figure 10. The F-statistic from all groups was bootstrapped resulting in a p-value of 0.0059. A post-hoc resampling statistical test was performed to compare the distributions of each groups in a one-to-one manner (see Methods). Color code: increase (green), decrease (red), and no significant changes (black) in the F1-score.

Supplementary Note 1: Modified MNIST Dataset

Parameters	Values
Epochs	150
Batch size	64
Objective function	Binary cross entropy with logits
Learning rate	0.01
Learning rate scheduler	N/A
Minimal learning rate	N/A
Overfitting criterion	N/A
Data augmentation	N/A

Supplementary Table 12: Training parameters of MICRA-Net for the modified MNIST dataset.

1.1 MICRA-Net segmentation and evaluation

To obtain the coarse segmentation, the procedure detailed in section 4.1 was followed. For each detected digits, a minimal certainty of 30% (qualitatively chosen from the validation set) was imposed to lower the rate of false-positive detection during segmentation and the `argmax` projection was used over the 10 classes to avoid overlap between detections. The feature maps was extracted from the network using the procedure explained in section 4.1 (Figure 2a-c). Precise binary segmentation maps were generated by applying a local Otsu threshold [2] for each 8×8 patch from the extracted coarse segmented region, which resulted in more accurate segmentation maps (Figure 2d,e & Supplementary Fig. 1). A bounding box of 28 pixels centered on each digit was used to compute the metrics, allowing a more balanced ratio between foreground and background pixels. The evaluated metrics are shown in Figure 2e & Supplementary Fig. 1.

1.2 Baseline architecture

For each network (fully-supervised and $\{5, 10, 25\}$ dilation dataset), the U-Net architecture was trained from scratch, keeping all hyper-parameters constant in all instances. U-Net architecture: Each step in the contracting path consists of two sets of 3×3 convolutional layers, followed by a batch normalization, and a 2×2 max-pooling. The number of filters in each layers was doubled after each contraction and are of size $\{16, 32, 64, 128\}$. The expanding path was symmetrical to the contracting path, but a 2×2 transposed convolution (stride of 2) was used to increase the layer size. Skip links are used to propagate information from higher layers. A final 1×1 convolutional layer was used to output the 11 classes segmentation map. The maximal argument along the class axis of the output was used as the semantic segmentation of the input image. ReLU activation was used throughout the network.

1.3 Baseline training procedure

The U-Net was trained from scratch using the ADAM optimizer with default values and with a learning rate 0.001 for 150 epochs. Binary cross entropy with logits was used as the loss to minimize. The model generalizing the most on the validation dataset was kept for testing. The U-Net was trained to output segmentation of 11 classes (all digits and background) resulting in a semantic segmentation of the images in the modified MNIST dataset. Since this architecture requires ground truth segmentation, the digits were used as binary masks to train U-Net in a fully-supervised manner (Figure 2f & Supplementary Fig. 2). This fully-supervised training consists in a high-standard baseline since U-Net has access to information that is not available in a weakly-supervised setting. For a more realistic baseline, the U-Net was also trained with weak annotations. To this end, the binary ground truth contours from the training dataset were dilated with an increasing size of the square structuring element ($\{5, 10, 25\}$ pixel), simulating an annotator contouring the images with coarse annotations, while still evaluating against undisturbed images (Figure 2f & Supplementary Fig. 2).

Supplementary Note 2: F-actin Dataset

Parameters	Values
Epochs	250
Batch size	32
Objective function	Binary cross entropy with logits with weights
Learning rate	0.001
Learning rate scheduler	- Reduce factor : 10 - Validation reduction : <0.01 - Patience : 10 epochs
Minimal learning rate	1×10^{-5}
Overfitting criterion	- Difference (validation - train) : 0.1 - Patience : 10 epochs
Data augmentation	- Horizontal flip - Vertical flip - Intensity scale - Gamma adaptation

Supplementary Table 13: Training parameters of MICRA-Net for the F-Actin dataset.

2.1 MICRA-Net weighted objective function

To account for data imbalance, a weighted binary cross entropy with logits was used:

$$l(\hat{y}, y) = L = \{l_1, ..., l_N\}^\top, l_n = p_n y_n \cdot \log \sigma(\hat{y}_n) + (1 - y_n) \cdot \log \sigma(1 - \hat{y}_n), \quad (1)$$

where p_n is the positive weight associated with class n. The positive weights during training were set to 3.3 and 1.6 for the periodical lattice and fibers respectively.

2.2 MICRA-Net segmentation and evaluation

The extracted feature map was globally thresholded to the 80^{th} percentile of the intensity of the image following a Gaussian blur with $\sigma = 1$. These parameters were selected based on a qualitative evaluation on the validation dataset. Data imbalance (between the number of foreground and background pixels) was accounted for by calculating the metrics inside the dendritic mask therefore reducing the number of negative pixels

2.3 Baseline architecture

The baseline architecture is a weakly-supervised U-Net trained using polygonal bounding boxes as annotations. The architecture is the same as in Lavoie-Cardinal et al. [3]. It is trained to output two independent segmentation maps, *i.e.* periodical lattice and fibers.

2.4 Baseline training procedure and evaluation

The baseline was trained as in Lavoie-Cardinal et al. [3]. The proper thresholds to generate binary segmentation maps for both structures was extracted from a receiver operating characteristic (ROC) curve on the validation dataset. As for MICRA-Net, the segmentation metrics were calculated inside the dendritic mask.

Parameters	Values
Epochs	700
Batch size	48
Objective function	Binary cross entropy with logits
Learning rate	1×10^{-4}
Learning rate scheduler	- Reduce factor : 2 - Validation reduction : <0.01 - Patience : 20 epochs
Minimal learning rate	1×10^{-5}
Overfitting criterion	- Difference (validation - train) : 1 - Patience : 10 epochs
Data augmentation	- Horizontal flip - Vertical flip - Random rotations (0-360°) - Intensity scale - Gamma adaptation - Shearing

Supplementary Table 14: Training parameters of MICRA-Net on the Cell Tracking Challenge dataset.

Supplementary Note 3: Cell Tracking Challenge Dataset

3.1 MICRA-Net training procedure

The number of generated crops from each cell line can greatly vary which could introduce imbalance bias between cell lines. On the other hand, the number of annotated images is more balanced than the number of generated crops from each cell lines. Hence, at the beginning of each epoch between 2 and 7 crops were randomly sampled from each annotated image in the dataset thereby creating a subset of crops that is balanced in terms of cell lines. The sampled subset of crops were randomly assigned to a mini-batch for optimization.

3.2 MICRA-Net segmentation and detection

MICRA-Net was trained on two different version of the dataset: i) 256×256 pixel, and ii) 128×128 pixel. For the former, we computed a dense prediction of the whole image. To avoid undesired edge effects, we padded each testing image using a symmetric padding and removed a 10 pixel border from the prediction. Hence, the pixel step between each predictions was 236 in both direction. We applied a Gaussian blur on the extracted feature map and used Otsu thresholding [2] to generate the segmentation mask. For latter, we used the fully-convolutional properties of MICRA-Net and computed a dense prediction using 1024×1024 pixel. Again, to avoid undesired edge effects we padded and cropped the border of the prediction, resulting in a pixel step of 1004 pixels. We computed the dense feature map of each testing images. We used Otsu thresholding for DIC-C2DH-HeLa, Fluo-N2DL-HeLa, PhC-C2DH-U373, and PhC-C2DL-PSC and used triangle thresholding for Fluo-C2DL-MSC and Fluo-N2DH-GOWT1. We noted that for the Fluo-N2DL-HeLa cell line, the network extracted edges of cells instead of the cell itself. For this reason, we inverted the generated segmentation map. We applied post-processing on the generated masks. We filled small holes ($< 100 \times 100$ pixel), removed small objects ($< 25 \times 25$ pixel), and applied binary erosion. We used a watershed algorithm on the resultant segmentation to split merged cells [4] followed by 10 iterations of consecutive binary erosion and dilation with a disk structuring element for smoothing. All operations were qualitatively assessed on the validation dataset.

To obtain the localization of detected objects, we used the centroid of each detected cells in each image. At inference, we used a maximal distance of 50 pixels for MICRA-Net to associate predicted objects with ground truth objects.

3.3 Baseline architecture

We used the pre-trained fully-supervised U-Net provided by Falk et al. [1]. This implementation of U-Net was pre-trained on the Cell Tracking Challenge dataset and other in-house datasets to differentiate between the cells and the background. We first converted the Caffe model in PyTorch. We essentially used the original architecture [1], but with 0-padding in the convolutional steps in order to output a segmentation map that has the same shape as the input image. To maintain similar classification task between MICRA-Net and U-Net we used 7 channels (6 cell lines and background) from which we take the `argmax` to generate a class specific segmentation.

3.4 Baseline training procedure and evaluation

For training, we used the Adam learning optimizer with a learning rate of 1×10^{-4} and default parameters. We reduced the learning rate by a factor of 2 when the validation did not reduce by more than 0.005 for the past 50 epochs, but kept a minimal learning rate of 1×10^{-5}. We used early stopping if the difference between the validation and training loss was higher than 1 for the last 10 epochs. We kept the model with the best generalization properties on the validation dataset for testing. A weighted cross-entropy loss was used as in Falk et al. [1] to help with the separation of cells. Briefly, we increased the weights between adjacent cells and increased the weights on cells compared to background. We trained the network for 1000 epochs with a batch size of 16. The size of the input crops were set at 256×256 pixel and the pixel step was set at 192 pixels. Similarly to MICRA-Net, we did not provide all crops in a single epoch as it introduced biases towards cell lines with more crops. We used the same data augmentation procedure than MICRA-Net.

At testing phase, we generated a dense prediction of the network with a sliding window of 256×256 pixel and a pixel step of 256 pixels. The `argmax` generated the semantic segmentation map. As in Falk et al. [1] we did not use any post-processing of the generated segmentation maps since the U-Net was trained with an objective function that i) encourages filled structures, and ii) increases the weights between adjacent cells to effectively split merged cells. The localization of each detected cells was obtained by calculating the centroid of the binary masks. We used a maximal distance of 30 pixels to calculate the association of objects.

Supplementary Note 4: P. vivax Dataset

Parameters	Values
Epochs	1200
Batch size	64
Objective function	Binary cross entropy with logits
Learning rate	1×10^{-4}
Learning rate scheduler	- Reduce factor : 2 - Validation reduction : <0.005 - Patience : 50 epochs
Minimal learning rate	1×10^{-5}
Overfitting criterion	- Difference (validation - train) : 1 - Patience : 10 epochs
Data augmentation	- Horizontal flip - Vertical flip - Random rotations (0-360°) - Intensity scale - Shearing

Supplementary Table 15: Training parameters of MICRA-Net on the P. vivax dataset.

4.1 MICRA-Net training procedure

The dataset was highly unbalanced towards uninfected red blood cells. Hence, we adapted the training procedure to balance the number of positive (infected red blood cells) and negative (background or uninfected cells) crops. Instead of training with all crops from an image at each epoch, we randomly sampled between 5 and 8 crops per images. The sampled crops were then randomly assigned to a mini-batch. We balanced the number of positive and negative crops by sampling an uninfected cell (or background) with a probability of 10%.

4.2 MICRA-Net segmentation

We applied a Gaussian blur with a sigma parameter of 3 and thresholded the resultant feature map at the 80^{th} percentile. We used small post-processing operations on the segmentation maps. We removed small holes ($< 50 \times 50$ pixel) from the generated binary masks and kept the most prominent object in cases where multiple infected cells were present in the extracted crops. This was necessary to evaluate the performance only on the subset of cells that were precisely annotated to generate the testing set. To do so, we selected the object which contained the maximal intensity from the extracted feature map of MICRA-Net. This procedure was qualitatively evaluated on the validation set.

4.3 MICRA-Net detection

We used a sliding window of size 256×256 pixel with a 32 pixels step and assigned the sigmoid transformed value to a 256×256 pixel crop in the image. We then calculated the average prediction at each pixel of the image and located all maxima using the `peak_local_max` function from the Scikit-Image Python library [5]. From these predicted positions and their associated probability, we generated a precision-recall curve to optimize the detection level of a given model on the validation set. We used a maximal distance of 128 pixels to associate detected and ground truth objects.

4.4 MICRA-Net fine-tuning procedure

To fine-tune MICRA-Net, we randomly selected $\{12, 24, 36\}$ images from the testing dataset. It is important to note that the 12 images were a subset of the 24 images, and the 24 images a subset of the 36 images (see Supplementary Fig 8). We tested several approaches to fine-tune MICRA-Net and compared their results on the resultant testing set (84 images). The approaches included i) adjusting the detection threshold [*Threshold*], fine-tuning the ii) linear layer [*Linear*] iii) linear layer and depth 4 [*Linear + 4*] iv) linear layer and depths 3 and 4 [*Linear + 3, 4*], and v) all [*All*] layers. To adjust the detection threshold, we used the 5 trained models and adjusted the detection threshold on the $\{12, 24, 36\}$ sampled images. For fine-tuning, we used a 3-fold training procedure to generate a small validation set for early stopping and to reduce the probability of over-fitting of the model (see Supplementary Fig. 8). We used Adam as the optimizer with a learning rate of 1×10^{-6} and trained for 100 epochs. We kept the state of the model generalizing the most on the validation set for each fold. The objective function used was binary cross entropy with logits. We did not reduce the learning rate of the models as in the original training phase. At inference, we used an ensemble model of the 3 trained models from each fold and averaged their predictions. We repeated the 3-fold training 5 times from each of the 5 naive models as base model, generating a total of 25 ensemble models (see Supplementary Fig. 8). We applied the same procedure for the detection and segmentation task as described above.

Supplementary Note 5: Scanning Electron Microscopy Dataset

5.1 MICRA-Net training procedure for SEM segmentation

We compared different Positive-Unlabeled (PU) ratios [6] for training $\{2:1, 1:1, 1:2, 1:5, 1:8, 1:16\}$. We randomly assigned negative crops according to the PU ratio. The negative crops of a specified PU ratio are a subset of the next higher PU ratio. Since, the extracted crops are larger in size (1024×1024 pixel)

Parameters	Values
Epochs	600
Batch size	16
Learning rate	1×10^{-4}
Objective function	Binary cross entropy with logits
Learning rate scheduler	- Reduce factor : 2 - Epochs : 50, 200, 300
Minimal learning rate	1×10^{-5}
Overfitting criterion	- Difference (validation - train) : 1 - Patience : 10 epochs
Data augmentation	- Horizontal flip - Vertical flip - Random rotations (0-360°) - Intensity scale - Gamma adaptation - Shearing - Elastic transform - Random position

Supplementary Table 16: Training parameters of MICRA-Net on the Scanning Electron Microscopy dataset.

than the training size (512×512 pixel), we could use random position sampling. This method increased the effective number of crops and served as another data augmentation technique.

5.2 MICRA-Net Segmentation

We applied a Gaussian blur with a sigma parameter of 5 and thresholded the resultant feature map at the 90^{th} percentile. We used small post-processing operations on the segmentation maps. We removed small holes ($< 45 \times 45$ pixel) from the generated binary masks and kept only the most prominent object for the same reason explained in P. vivax dataset. To do so, we selected the object which contained the maximal intensity from the generated feature map of MICRA-Net. This procedure was qualitatively evaluated on the validation set.

5.3 MICRA-Net Detection

A sliding window of size 512×512 pixel with a 128 pixels step was used to assign the sigmoid transformed value to a 512×512 pixel crop. We averaged the predictions at each pixel of the image and located all maxima using the `peak_local_max` function from the Scikit-Image Python library [5]. Using these predicted positions and their associated probability, we could generate a precision-recall curve to optimize the detection level of a given model on the validation set. We used a maximal distance of 512 pixels to associate detected and ground truth objects.

References

[1] Thorsten Falk, Dominic Mai, Robert Bensch, Özgün Çiçek, Ahmed Abdulkadir, Yassine Marrakchi, Anton Böhm, Jan Deubner, Zoe Jäckel, Katharina Seiwald, et al. U-net: deep learning for cell counting, detection, and morphometry. *Nature Methods*, 16(1):67, 2019.

[2] Nobuyuki Otsu. A threshold selection method from gray-level histograms. *IEEE Transactions on Systems, Man, and Cybernetics*, 9(1):62–66, 1979.

[3] Flavie Lavoie-Cardinal, Anthony Bilodeau, Mado Lemieux, Marc-André Gardner, Theresa Wiesner, Gabrielle Laramée, Christian Gagné, and Paul De Koninck. Neuronal activity remodels the f-actin based

submembrane lattice in dendrites but not axons of hippocampal neurons. *Scientific Reports (Nature Publisher Group)*, 10(1), 2020.

[4] Serge Beucher and Fernand Meyer. The morphological approach to segmentation: the watershed transformation. *Mathematical morphology in image processing*, 34:433–481, 1993.

[5] Stefan Van der Walt, Johannes L Schönberger, Juan Nunez-Iglesias, François Boulogne, Joshua D Warner, Neil Yager, Emmanuelle Gouillart, and Tony Yu. scikit-image: image processing in python. *PeerJ*, 2: e453, 2014.

[6] Jessa Bekker and Jesse Davis. Learning from positive and unlabeled data: a survey. *Mach. Learn.*, 109 (4):719–760, 2020.