A Deep Learning Pipeline for Nucleus Segmentation

George Zaki[1*§], Prabhakar R. Gudla[2*], Kyunghun Lee[2], Justin Kim[1#], Laurent Ozbun[2], Sigal Shachar[3], Manasi Gadkari[4], Jing Sun[5], Iain D.C. Fraser[5], Luis M. Franco[4], Tom Misteli[3], and Gianluca Pegoraro[2§].

1: Biomedical Informatics and Data Science Directorate, Frederick National Laboratory for Cancer Research (FNLCR), Frederick, MD, 21702.
2: High-Throughput Imaging Facility (HiTIF), Center for Cancer Research (CCR), NCI/NIH, Bethesda, MD, 20892.
3: Cell Biology of Genomes (CBGE), Center for Cancer Research (CCR), NCI/NIH, Bethesda, MD, 20892.
4: Systemic Autoimmunity Branch, NIAMS/NIH, Bethesda, MD, 20892
5: Laboratory of Immune System Biology, NIAID/NIH, Bethesda, MD, 20892.
#: Present Address, Pratt School of Engineering, Duke University, Durham, NC, 27708.

*: These authors contributed equally to this work
§: Corresponding authors

Running headline: Practical Deep Learning for Nucleus Segmentation

Corresponding Authors Information
Gianluca Pegoraro
gianluca.pegoraro@nih.gov

George Zaki
george.zaki@nih.gov

Abstract

Deep learning is rapidly becoming the technique of choice for automated segmentation of nuclei in biological image analysis workflows. In order to improve and understand the training parameters that drive the performance of deep learning models trained on small, custom annotated image datasets, we have designed a computational pipeline to systematically test different nuclear segmentation model architectures and model training strategies. Using this approach, we demonstrate that transfer learning and tuning of training parameters, such as the training image dataset composition, size and pre-processing, can lead to robust nuclear segmentation models, which match, and often exceed, the performance of existing, state-of-the-art deep learning models pre-trained on large image datasets. Our work provides computational tools and a practical framework for the improvement of deep learning-based biological image segmentation using small annotated image datasets.

Key terms: Deep Learning, Machine Learning, Nucleus Segmentation, Fluorescence Microscopy, High-Content Imaging

Introduction

High-Content Imaging (HCI) uses automated liquid handling, image acquisition and image analysis to screen the biological effect of hundreds of thousands of perturbing agents, such as RNAi, CRISPR/Cas9, and chemical compounds, by measuring cellular phenotypic changes in microscopy-based assays of interest (1,2). The precise and automated nuclear segmentation of up to hundreds of thousands of fluorescence microscopy images of nuclei stained with specific fluorescent dyes is often the first essential analysis step in HCI (3), because most cell types possess only one centrally positioned nucleus and the nucleus is used to identify individual cells. In addition, nuclei tend to have a regular shape, and can be easily visualized with fluorescent DNA stains, which are cheap and provide high contrast. Nevertheless, notable exceptions to these desirable properties exist: cell lines and primary cells can have dramatically different nuclear sizes, there can be substantial differences in nuclear shape (such as lobulation in polymorphonuclear cells of the immune system), and cells can grow tightly packed and/or in colonies thus leading to adjacent or overlapping nuclei. This makes it challenging to develop, test and implement robust algorithms that can provide accurate and automated nuclear segmentation results for a variety of cell types and in different experimental conditions.

In recent years, computational advances in deep convolutional neural-networks (CNNs) have produced models that match, and sometimes exceed, human levels of performance in a variety of classification, clustering, and segmentation tasks in biological image analysis (4–6). More specifically, several studies have described novel CNN-based approaches for the segmentation of nuclei from fluorescence microscopy images (7–13). These efforts were mostly focused on training a variety of CNN architectures with fairly large and open source fluorescent image datasets of cells and nuclei, with the goal of ensuring that trained models could generalize on novel image datasets without the need to retrain them (11,12). While the collection and distribution of pre-trained models and large datasets of annotated biological images is highly useful, less is known about the performance of pre-trained models on images of primary cells or cell lines which possess

nuclear shapes previously not "seen" by the pre-trained models, and which were acquired on different microscopes (9,14). Also, still much needs to be learned about the feasibility of, and the necessary steps for, obtaining precise nuclear segmentation by training CNN models on relatively small (500 – 1,000 nuclei) sets of images, which are readily available in most laboratories (9,13,14). Finally, from an end-user standpoint, few guidelines exist on how pre-existing nuclear segmentation models can be trained, adapted and improved on new image datasets (13).

To address these questions, we designed and implemented an end-to-end computational pipeline to quickly train and evaluate the performance of deep –learning-based nuclear segmentation algorithms. This pipeline was first used to generate preliminary nuclear labels from a few images, which were then manually corrected in an interactive fashion, and finally used to train and evaluate the performance of different CNN-based architectures for nuclear segmentation. The pipeline was used to experiment with different training strategies for two pre-existing CNN-based image segmentation model architectures: Feature Pyramid Network2-Watershed (FPN2-WS) (15), and Mask R-CNN (MRCNN) (16). To mimic practical scenarios in the laboratory, we trained and applied these two models to segment nuclei in four cell types that have very different nuclear morphology, and that were acquired at different magnifications. Our results describe a practical path for researchers to train different CNN architectures in a semi-automated manner, and indicate that these deep learning models can be efficiently trained with a small number of images to segment heteromorphic nuclei from microscopy images.

Materials and Methods

Cells

HCT-116 and U2OS cells were obtained from ATCC (Mansassas, VA, USA). MCF10A cells were a kind gift of Daniel Haber (Harvard University, MA, USA). Cells were grown in 384-well plates (CellCarrier Ultra 384, PerkinElmer, Waltham, MA), fixed in 4% PFA and stained with DAPI. Primary human eosinophils were obtained from healthy donors enrolled under NIH protocol NCT000090662 and purified as previously described (17). The protocol was approved by the Institutional Review Board of the National Institute of Allergy and Infectious Diseases (NIAID) at the National Institutes of Health (NIH). Eosinophils were plated in Poly-D-Lysine coated 384-well plates (CellCarrier Ultra 384, PerkinElmer, Waltham, MA), then fixed in 4% PFA and stained with DAPI. Cells were plated and stained with DAPI either in biological replicate wells (i.e., cells were plated on different plates on different days) for MCF10A and HCT-116 cells, or technical replicate wells (i.e., cells were plated on the same plate, on the same day, but in different wells) for U2OS and primary eosinophils.

Image Generation

Fluorescence microscopy images of 384-well plates were acquired on a CV7000S spinning disk confocal microscope (Yokogawa, Japan). For all cells, we used a 405 nm solid-state excitation laser, a 405/488/568/640 nm quad excitation dichroic mirror, a 568 nm emission dichroic mirror, a 445/45 nm bandpass emission mirror, and an sCMOS camera (2550 X 2160 pixels). For MCF10A and eosinophils, we used a 60X water objective (NA 1.2), z-sectioning every 0.5 microns, and camera binning set to 2X2 pixels. For U2OS cells we used a single z-plane acquisition with a 20X air objective (NA 0.75) and camera binning set to 1X1 pixel. For HCT-116 cells we used a single z-plane acquisition with a 20X air objective (NA 0.75) and camera binning set to 2X2 pixels. Proprietary geometric and shading correction software was run on the fly during acquisition for all cell types. In the case of 3D image acquisition, images were maximally projected on the fly.

Computational Pipeline

The computational pipeline was based on the Snakemake language (18), and developed as a reproducible workflow capable of evaluating any number of different nuclear instance segmentation models. The analysis workflow was designed to run either on high-performance computing (HPC) clusters or on a local server equipped with Graphics Processing Units (GPUs). Snakemake is a generic workflow management software, where individual computational tasks are chained together. Each task is configured by a set of input files and parameters, by a call to a bash or a Python script that accepts the input files and processes them, and by a set of output files generated by the task.

Ground Truth Labels Generation

The nucleus labels used in supervised learning were generated in two steps. First, preliminary labels were automatically generated using either classical image processing techniques, or existing, non-trained nuclear segmentation deep learning models. In particular, since nuclei of MCF10A cells displayed high contrast and good separation, ground truth labels (instance segmentation masks) for these cells were generated using a simple nuclear segmentation pipeline in KNIME (19,20), which included thresholding, gaps filling, connected components, and manual separation of a few adjacent nuclei. A set of Python scripts was used to convert these preliminary nuclear labels images into a format compatible with the interactive, web-based image editing tool Supervisely (21). An expert cell biologist corrected nuclear segmentation mistakes using a combination of bitmask brushes and polygons to generate a set of high-quality ground truth nuclear labels. The correction task took about 2 hours for 1,000 nuclei.

The nuclear ground truth labels for MCF10A cells were used for a first round of supervised training for the two deep learning models used in this work, MRCNN and FPN2-WS (See below for details). The models trained in this fashion were then used in

inference mode to generate preliminary labels for all other cell types, which were then similarly corrected using Supervisely to obtain ground truth labels.

Image Augmentation

Image augmentation of full fields of view (FOVs) was used to simulate slightly different illumination conditions, to enrich the training image dataset, and to avoid model overfitting. Multiple augmentation methods from the Python module imgaug (22) were used. In particular, our pipeline can be configured to use scaling transformations (0.33X, 0.66X, 1X, 1.33X, and 2X), affine transformations (vertical and horizontal flips, 45^0, 90^0, 135^0, 180^0, 225^0, 270^0, and 315^0 rotations), contrast transformations (Contrast Limited Adaptive Histogram Equalization (CLAHE), gamma correction, and saturation), blurring (median blur and gaussian blur), or random noise addition (impulse noise, gaussian noise, and dropout).

When only MCF10A cells were used for model training, 7,000 overlapping patches of size 256 X 256 px were randomly sampled from full FOVs of the grayscale images, and from the matching ground truth labels images. 5 scaling transformations were then applied to these patches, to obtain a total of 35,000 augmented patches. In cases when all four cell types were used for model training, each cell type contributed 8,750 scaled and augmented patches, again for a total of 35,000 patches. For each scaled patch one of the affine transformations was randomly picked and applied. For each scaled and rotated or flipped patch, one of the intensity augmentation methods mentioned above was randomly selected with a probability between 0 and 1, and applied to the grayscale patch and to the nuclear label patch, when appropriate. For the MRCNN architecture, the augmented ground truth labels were converted to binary mask images. For the FPN2-WS architecture, the labels were further transformed using traditional image processing operations (Contour generation, distance transform) to obtain the normalized distance transform and gaussian blurred grayscale images used in this model (See below for further details). Augmented patches from all the cell types were combined and saved in .h5 file format.

In experiments where the image augmentations were turned off, we assigned a probability of 0 to the augmentation classes to disable them. Rotations and Flips were applied in all training experiments.

Model Description

Mask R-CNN (MRCNN)

We used the Matterport implementation of Mask R-CNN (MRCNN) (23) with minor modifications. In particular, we added a parser to accept images in h5 format and set the size of the input layer for both model training and inference to 256 X 256 px. At inference time, full FOVs were first divided into grids of 256 X 256 px overlapping patches, where the overlap value was set to 50 pixels, which on average was enough to cover one nucleus. Model inference was then run on the patches for a single FOV to generate nuclear labels with a unique ID. Nuclear labels with a different ID that were overlapping above a certain threshold across two or more patches were then computationally merged into a nuclear label with a single ID. Nuclear labels with an area below a certain threshold were eliminated. Finally, since MRCNN can sometimes predict more than one connected component per nuclear label ID, only the largest connected component for every nuclear label ID was retained.

Feature Pyramids Network 2 - Watershed (FPN2-WS)

In the FPN2-WS architecture, two feature pyramids networks (FPN) models (24) were trained to predict a normalized distance transform image for each nucleus and a gaussian blurred version of the nuclear label border, respectively. The distance transform was then thresholded at a value of 0.6. The thresholded normalized distance transform and the blurred border images were used for watershed nuclear segmentation (25).

Model Training

For all 4 cell types, full FOV of DAPI-stained cells from a single replicate well were used as a training dataset. One training cycle was defined as a single pass of the entire

augmented dataset through the network. We decided to use the term "cycle", as opposed to the more traditional term "epoch", because the original images were randomly bootstrapped to generate a training dataset of 35,000 256 X 256 px patches. Unless otherwise stated, the MRCNN and FPN2-WS models were trained for a total of 25 cycles. For MRCNN, we used the same learning rate schedule as previously described in the MatterPort implementation of this model (23). In particular, since the MatterPort schedule is defined over 200 iterations, then we distributed the 25 training cycles over 200 iterations starting with a learning rate of 10e-4. The model with the lowest test loss value over 25 cycles was saved and used for inference. The model parameters are available on github (https://github.com/CBIIT/nci-hitif/tree/master/framework-nucleus-segmentation).

When ImageNet weights were used, the parameters of the encoder portion of the network were frozen during training for the first 2 cycles, then all trainable layers continued training for the remaining 23 cycles. We selected a sigmoid activation function, and the Adam optimizer to minimize the mean square error of the corresponding network output image (i.e., distance transform or blurred contour, respectively). The learning rate was set to a constant value of 1e-3.

Model Testing

For inference, each model was tested on images of nuclei from all cell types obtained either from technical (U2OS, eosinophils) or biological replicate wells (MCF10A, HCT-116) that were not used for model training.

Quantitative assessment of nuclear instance segmentation performance

Given ground truth nuclei IDs and the inference nuclei IDs, the F1 scores were calculated at different Intersection over Union (IoU) thresholds. A two-dimensional histogram was used to calculate the intersection area (i.e., the number of pixels) between all pairs of nuclei from the ground truth labels and the inference labels. The union area between

every pair of ground truth and inference labels was calculated as the sum of the pixels belonging to either of these labels minus their intersection. For every pair of ground truth and inference labels, the IoU was calculated as the ratio between the intersection over the union. For any IoU threshold t between 0 and 1, the True Positive ($TP(t)$) statistic was the number of ground truth labels that have an IoU value equal or higher than t with one inference label. Similarly, the False Negative ($FN(t)$) statistic was the number of ground truth labels that have an IoU value lower than t with any inference labels. Finally, the False Positive ($FP(t)$) was the number of inference labels that have an IoU value lower than t with any ground truth label.

Given $TP(t)$, $FN(t)$ and $FP(t)$, the F1 score at a given IoU threshold t was calculated as:

$$F1(t) = TP(t) / (TP(t) + (FP(t) + FN(t))/2)$$

Software repository

The Snakemake model training pipeline, the weights for the best MRCNN and FPN2-WS models, the images used, and the code used to generate Figs. 2– 5) can be found in the HITIF github repository, and in a dedicated Figshare project.

Results

We aimed to test the practical implementation of different CNN image segmentation models and training strategies for the automated identification of nuclei from a wide variety of fluorescence microscopy images. To this end, we built a computational pipeline (See Materials and Methods for details) to train the CNNs starting from a limited number of fluorescence microscopy images (~ 500 – 1,000 nuclei), which is a typical use case scenario for a single cell biologist/researcher (Fig. 1A). As a first step in the pipeline, a pre-trained CNN model or traditional nucleus segmentation approaches were used to generate preliminary sets of labels for the nuclei. Depending on a variety of factors, these preliminary labels contained gross segmentation mistakes, which had to be corrected using an interactive, web-based interface (Fig. 1A, and Materials and Methods). This high-quality set of manually annotated ground-truth labels was then used to train CNN models in combination with the original grayscale images of DAPI-stained nuclei (Fig. 1A and 1B). Finally, the computational pipeline measured the inference performance of previously trained segmentation models by automatically calculating F1 scores at different Intersection over Union (IoU) thresholds for the original ground-truth labels and the newly inferred labels (11) (Fig. 1A, and Materials and Methods for details).

Since we decided train the deep learning segmentation models from relatively small datasets of images, we decided to bootstrap the datasets by generating multiple random overlapping patches from the original full field of view (FOV), and to augment these patches by randomly applying several image transformations (Fig. 1B and Materials and Methods for details). Following these steps, the CNNs were trained with an equal number of augmented images/patches for each cell line (Fig. 1B) (22). In addition, to speed up training and to ensure that the models would reach convergence during training we also adopted a transfer learning approach (26,27), where the models were initialized with weights obtained from training the models on large datasets of everyday images, such as ImageNet (28) or COCO (29) (Fig. 1B).

To test the training pipeline for deep learning nuclear segmentation on different CNN models architectures, we trained two models widely used in instance segmentation tasks: Feature Piramid Network - 2 Watershed (FPN2-WS) (15), and Mask-RCNN (MRCNN) (16). In particular, and similarly to previously described approaches (11,30), FPN2-WS (Fig. 1C) consisted of two separate FPN models to predict a normalized distance transform image for the nucleus and a Gaussian blurred image of its border, respectively (Fig. 1C). These two images were then thresholded and used as the seed and border, respectively, for a traditional watershed segmentation algorithm to obtain the final nuclear labels (Fig. 1C).

Our baseline consisted of randomly initialized MRCNN and FPN2-WS models, trained using only 7 FOVs of nuclei from the epithelial MCF10A cell line, which has a regular nuclear morphology and, for the most part, does not form clumps or occlusions (Fig. 2A). We then tested the inference performance of the trained MRCNN and FPN2-WS models on previously unseen images of MCF10A nuclei from another biological replicate imaged at 60X magnification, on the osteosarcoma cell line U2OS imaged at 20X, on the colorectal cell line HCT-116 imaged at 20X, and on primary human eosinophils imaged at 60X (Fig. 2A). These cell types were chosen to test the performance of the trained nuclear segmentation architectures on a range of different nuclear shapes, sizes, confluency, and occlusions (Fig. 2A). Not surprisingly, visual comparison of the Ground Truth nuclear labels predicted by the MRCNN and FPN2-WS trained as indicated above (Fig. 2A, GT panels), with the nuclear labels predicted by the models (Fig. 2A, Inference panels), indicated the presence of several prediction errors (Fig. 2A, GT XOR Inference panels). These included instances of undersplitting, oversplitting, missed nuclei, and mistakenly drawn nuclear borders. Errors were particularly evident in HCT-116 cells, which tend to grow in clumps, and in primary eosinophils, which have a highly lobulated nucleus (Fig. 2A). Accordingly, quantification of these images by measuring F1 inference scores (See Materials and Methods for details) at an IoU threshold of 0.7 (11), showed that MRCNN had F1 score values of 0.82, 0.84, 0.63, and 0.72 on MCF10A, U2OS, HCT-116, and primary eosinophils, respectively (Fig. 2B), while the FPN2-WS model had F1 values of 0.97, 0.94,

0.84, and 0.48 on MCF10A, U2OS, HCT-116, and primary eosinophils, respectively (Fig. 2B). The results of this first set of experiments suggest that training CNN models with random initialization using a single cell type does provide, at least in the case of FPN2-WS, good nuclear segmentation performance on the same cell type, or on cells with similar nuclear morphology and confluency characteristics (Fig. 2A, Compare MCF10A with U2OS). At the same time, the models failed to generalize at inference time on other cell types with more heteromorphic nuclei or growth characteristics (Fig. 2A, Compare MCF10A with HCT-116 and primary eosinophils).

We thus tested whether changing the model training strategy would improve the nuclear segmentation performance of these deep learning models on our image inference dataset. In particular, to avoid an excessive increase in the number of values and parameters to be tested, we decided to not change CNN network architectures, the training optimizing algorithms, or the learning rate constant. Instead, we focused on varying the network's weights initialization (i.e., transfer learning), the composition of the image training dataset, the number of training cycles, the size of the training dataset, and on the augmentation strategy. We began testing the effect of these changes by initializing the MRCNN and FPN2-WS models with weights obtained by training them on the COCO or ImageNet datasets, respectively (Fig. 3A). As compared to random initialization, pre-initialization with COCO weights and transfer learning increased F1 inference scores of the MRCNN model on the MCF10A (0.82 vs. 0.97), U2OS (0.84 vs. 0.97) and HCT-116 (0.63 vs. 0.88) cell lines, while leading to a decrease (0.72 vs. 0.63) for the primary eosinophils. In similar experiments using the FPN2-WS model, transfer learning had a marginal increase of F1 scores on MCF10A, U2OS and HCT-116 (0.97 vs. 0.98, 0.94 vs. 0.96, and 0.84 vs. 0.85, respectively), and a moderate increase of the F1 score in eosinophils (0.48 vs. 0.58) (Fig. 3A). Having observed a positive effect of transfer learning on nuclear segmentation tasks using these models, we then progressively increased the variety of images used in training of pre-initialized models to test whether these are capable of learning a set of nuclear segmentation features for multiple cell types in sequential or simultaneous fashion (Fig. 3B). The F1 scores for MCF10A, U2OS and HCT-116 nuclei did

not appreciably increase or decrease upon training MRCNN and FPN2-WS with data from multiple cell types (Fig. 3B). On the other hand, for the primary eosinophils, the F1 score for MRCNN increased when images from these cells were used in training, as compared to when only MCF10A cells were used (Fig. 3B, 0.63 vs. 0.80). These results indicate that using pre-initialization of models with weights from large image datasets has a positive effect on nuclear segmentation using the MRCNN model on most types tested. In addition, sequential training of deep learning models with increasingly varied image training sets does not have a negative impact on the performance of the deep learning models tested here, and in the case of primary eosinophils, which are polymorphonuclear cells, it can improve the performance of MRCNN.

Deep learning model training and image pre-processing steps, such as image augmentation, often require substantial amounts of time and computing power. We thus considered the impact of reducing the number of training cycles (Fig. 4A), or the size of the image training datasets for pre-initialized MRCNN or FPN2-WS models trained on images from all the cell types (Fig. 4B). Reducing the number of training cycles by a factor of 4 (Fig. 4A, 1X vs 0.25X) lead to a slight decrease in F1 inference scores for MCF10A (MRCNN: 0.97 vs. 0.93, FPN2-WS: 0.98 vs. 0.98), U2OS (MRCNN: 0.94 vs 0.91, FPN2-WS: 0.97 vs. 0.95), and HCT-116 (MRCNN: 0.89 vs. 0.87, FPN2-WS: 0.87 vs. 0.85), while it led to a larger decrease for primary eosinophils (MRCNN: 0.80 vs. 0.76, FPN2-WS: 0.46 vs. 0.35). Similarly, to test whether we could maintain the same F1 inference scores by training the models with less data, or if we could improve them by larger training datasets, we trained pre-initialized MRCNN and FPN2-WS with image datasets that included all the cell types reflecting a 16-fold range of sizes (Fig. 4B, 0.25X to 2X) as compared to the dataset used in the previous experiments (1X, 8,750 augmented patches per cell type) (Fig. 4B). Training with twice the amount of data did not lead to an improvement in F1 inference scores for either model in any of the cell types (Fig. 4B). Reducing the amount of data 4-fold from the original dataset instead had more diverse effects on the F1 inference scores, ranging from negligible (e.g. Fig 4B, MCF10A, MRCNN, 0.97 vs. 0.96) to substantial decreases (e.g. Fig 4B, eosinophils, MRCNN, 0.80 vs. 0.53).

Finally, we tested the effect of eliminating one class of augmentation operation at a time, or all at the same time, to test the contribution of each class to the performance of the model (Fig. 4C). Surprisingly, for both MRCNN and FPN2-WS, and as compared to implementing the full set of augmentations, skipping all image augmentations did not degrade the inference performance of the model in any cell type (Fig. 4C) and actually slightly improved F1 inference scores for MRCNN in the case of U2OS cells (0.94 vs 0.97), HCT-116 (0.90 vs 0.91) and primary eosinophils (0.80 vs. 0.82). Having settled on a refined set of model training parameters, we then decided to run replicate training experiments with these to ensure that the model training regimen was stable and to estimate the variance in inference performance between runs (Fig 4D). The results of this set of replicate experiments confirm that training of these models leads to reproducible F1 inference performance, and that initialization with pre-trained weights leads to a significant improvement in F1 inference scores with MRCNN and FPN2-WS for several cell types (Fig. 4D). In addition, they also indicated that skipping image augmentation before MRCNN training significantly improved F1 scores for HCT116 and primary eosinophils, without leading to a significant drop in performance for MCF10A or U2OS cells (Fig. 4D). These results indicate that reducing the number of training cycles and datasets, and skipping image augmentation, are viable steps in the training of these models, without leading to a reduction in nuclear segmentation performance.

Having settled on a refined set of model training parameters for MRCNN and FPN2-WS, we then benchmarked the nuclear segmentation inference performance of these models against a pre-trained, state-of-the-art Jacobkie deep learning model for nuclear segmentation (30). Jacobkie has a similar model architecture as FPN2-WS, was trained on a large and diverse dataset of images of nuclei from different sources, won 2nd place out of about 10,000 submissions to the 2018 Kaggle Data Science Competition, and it has been suggested that this model can be used in inference without pre-training (11). Visual comparison of inferred nuclear segmentation masks indicated that, while Jackobie was generally less precise than MRCNN or FPN2-WS at the pixel segmentation level, it achieved a comparable instance segmentation performance for MCF10A and HCT-116

cells, and an inferior performance on U2OS or primary eosinophils (Fig. 5A). In particular, the MRCNN segmentation model seemed to better generalize to irregular nuclear shapes, such as those of primary eosinophils (Fig. 5A). In line with the qualitative visual results, comparison of F1 segmentation scores for the three segmentation models indicated that MRCNN and FPN2-WS achieved similar or better F1 inference scores than Jacobkie for all cell types both at an IoU of 0.9 (Fig. 5B), which is indicative of segmentation precision at the pixel level, and at an IoU of 0.7 (Fig. 5B), which is indicative of instance segmentation precision. Overall, these results confirm that, while in some cases state-of-the-art nuclear segmentation models, such as Jacobkie, can be used for nuclear instance segmentation in inference mode without retraining, optimal nuclear segmentation performance on a variety of cells likely requires *de novo* training of CNN models using a small number of annotated images.

Discussion

Machine learning approaches have found wide applications in HCI because they can learn rapidly from the datasets and they generate close to optimal, and parameter-free solutions to a wide variety of previously non-addressable image processing tasks (6). In particular, in supervised CNN-based machine learning tasks one of the main challenges is to choose appropriate combinations of suitable machine learning model architectures, image pre-processing or post-processing strategies, and model training regimens. To automate the training and testing of a large set of these combinations, we designed an end-to-end computational pipeline that includes ground-truth label generation, image pre-processing, model training, image post-processing, and quantitative model performance evaluation steps. The pipeline is based on Snakemake, a widely used open-source pipeline software (18) that can be run either locally, or on an HPC cluster equipped with one or more GPUs (See Materials and Methods for details).

We used the automated computational pipeline to run a total of 76 supervised model training experiments for CNNs tasked to perform segmentation of cell nuclei from fluorescence microscopy images of fixed cells stained with the DNA stain DAPI. To test the usage of nuclear segmentation algorithms in a typical heterogenous laboratory setting, we ran training and inference experiments on a panel of 4 cell types with substantial differences in image acquisition conditions (i.e., magnification and camera binning), nucleus size, nucleus morphology, and degree of confluency. Overall, the FPN2-WS model performed extremely well on 2 of the 4 cell types tested, with or without pre-initialization, and just by training it with images from one cell type (Fig. 2A). On the other hand, while MRCNN did not perform as well as FPN2-WS in these starting conditions, it showed comparable or better performance on most cell types when transfer-learning (26) and a training set containing images of the 4 cell types were adopted (Fig. 3A and 3B). Paradoxically, in the case of MRCNN, adding image augmentation or increasing the number of patches used per cell type slightly deteriorated the segmentation performance of the model. While we cannot exclude that these approaches might be useful for deep learning-based nuclear segmentation on images

acquired in other contexts, our results underscore the importance of testing different training parameters to identify the most important ones, and to potentially reduce the computational load and/or time required for model training. Future efforts to improve the performance of MRCNN on images of nuclei from different cell types, acquired on different fluorescence microscopes, with different microscopy modalities, or different sample types (e.g. histological tissue sections) could take advantage of more advanced transfer learning approaches such as model weight transfusion (27).

Furthermore, comparison of the performance of MRCNN and FPN2-WS trained on our dataset with Jacobkie, a pre-trained state-of-the-art model (11), indicates that using pre-trained models is an effective, low-cost trade-off on images of nuclei of certain cell types, but that training and testing additional models on the images of interest becomes necessary if the final goal is to obtain high pixel-level precision, or if the nuclei in question are substantially different from the ones present in the original training set, such as primary eosinophils or HCT-116 cells (Fig 5A and 5B). To this end, we suggest that end-users might first test the performance of a pre-trained nuclear segmentation model on the images of interest, and, if these results are not satisfactory, then edit the preliminary nuclear labels and apply transfer learning to the model to improve its performance.

We expect that the training pipeline described here, and this practical set of guidelines, will help researchers to test novel model architectures and novel image pre- and post-processing steps aimed at the segmentation of nuclei and other biological structures from microscopy images. In the future, we expect it will be feasible to expand this set of observations to 3D images of nuclei, and to the segmentation of other cellular organelles and sub-structures, such as mitochondria, vesicles, or nuclear bodies.

Acknowledgements

The authors thank the CBIIT Server Team, NCI, National Institutes of Health (NIH) and the High-Performance Computing Group, CIT, NIH for computational support. Also, we would like to thank Murat Bilgel for contributing to the MRCNN postprocessing

implementation, and Andrew Weisman for contributing to the image augmentation wrapper. We thank Amy Klion and Michelle Makiya at the National Institute of Allergy and Infectious Diseases for providing primary human eosinophils. Contributions: GZ, PRG and GP conceptualized and designed the project; GZ and JK refactored the code for the MRCNN model; PRG wrote the code for the FPN2-WS model; GZ, PRG, and KL wrote and executed the nuclear segmentation training pipeline; LO, SS, MG, JS, IF, LMF, TM, and GP provided the training images dataset; GZ and GP analyzed the results of the experiments; GZ and GP wrote the manuscript; all authors read and revised the manuscript. This research was supported by funding from the Intramural Research Program of the National Cancer Institute, Center for Cancer Research Project Num. 1ZICBC011567-01, the National Institute of Allergy and Infectious Diseases, and the National Institute of Arthritis and Musculoskeletal and Skin Diseases, all at the National Institutes of Health (NIH). All the authors declare no conflict of interest.

References

1. Boutros M, Heigwer F, Laufer C. Microscopy-Based High-Content Screening. Cell 2015;163:1314–1325.

2. Pegoraro G, Misteli T. High-Throughput Imaging for the Discovery of Cellular Mechanisms of Disease. Trends Genet. 2017;33:604–615.

3. Ljosa V, Carpenter AE. Introduction to the quantitative analysis of two-dimensional fluorescence microscopy images for cell-based screening. PLoS Comput. Biol. 2009;5:e1000603.

4. Kraus OZ, Grys BT, Ba J, Chong Y, Frey BJ, Boone C, Andrews BJ. Automated analysis of high-content microscopy data with deep learning. Mol. Syst. Biol. 2017;13:924.

5. Ching T, Himmelstein DS, Beaulieu-Jones BK, Kalinin AA, Do BT, Way GP, Ferrero E, Agapow P-M, Zietz M, Hoffman MM, Xie W, Rosen GL, Lengerich BJ, Israeli J, Lanchantin J, Woloszynek S, Carpenter AE, Shrikumar A, Xu J, Cofer EM, Lavender CA, Turaga SC, Alexandari AM, Lu Z, Harris DJ, DeCaprio D, Qi Y, Kundaje A, Peng Y, Wiley LK, Segler MHS, Boca SM, Swamidass SJ, Huang A, Gitter A, Greene CS. Opportunities and obstacles for deep learning in biology and medicine. J. R. Soc. Interface 2018;15. Available at: http://dx.doi.org/10.1098/rsif.2017.0387.

6. Moen E, Bannon D, Kudo T, Graf W, Covert M, Van Valen D. Deep learning for cellular image analysis. Nat. Methods 2019. Available at: http://dx.doi.org/10.1038/s41592-019-0403-1.

7. Kraus OZ, Ba JL, Frey BJ. Classifying and segmenting microscopy images with deep multiple instance learning. Bioinformatics 2016;32:i52–i59.

8. Sadanandan SK, Ranefall P, Le Guyader S, Wählby C. Automated Training of Deep Convolutional Neural Networks for Cell Segmentation. Sci. Rep. 2017;7:7860.

9. Caicedo JC, Roth J, Goodman A, Becker T, Karhohs KW, Broisin M, Molnar C, McQuin C, Singh S, Theis FJ, Carpenter AE. Evaluation of Deep Learning Strategies for Nucleus Segmentation in Fluorescence Images. Cytometry A 2019. Available at: http://dx.doi.org/10.1002/cyto.a.23863.

10. Schmidt U, Weigert M, Broaddus C, Myers G. Cell Detection with Star-convex Polygons. arXiv [cs.CV] 2018. Available at: http://arxiv.org/abs/1806.03535.

11. Caicedo JC, Goodman A, Karhohs KW, Cimini BA, Ackerman J, Haghighi M, Heng C, Becker T, Doan M, McQuin C, Rohban M, Singh S, Carpenter AE. Nucleus segmentation across imaging experiments: the 2018 Data Science Bowl. Nat. Methods 2019. Available at: http://dx.doi.org/10.1038/s41592-019-0612-7.

12. Stringer C, Michaelos M, Pachitariu M. Cellpose: a generalist algorithm for cellular segmentation. bioRxiv 2020:2020.02.02.931238. Available at: https://www.biorxiv.org/content/10.1101/2020.02.02.931238v1.full.pdf+html. Accessed February 27, 2020.

13. Van Valen DA, Kudo T, Lane KM, Macklin DN, Quach NT, DeFelice MM, Maayan I, Tanouchi

Y, Ashley EA, Covert MW. Deep Learning Automates the Quantitative Analysis of Individual Cells in Live-Cell Imaging Experiments. PLoS Comput. Biol. 2016;12:e1005177.

14. Falk T, Mai D, Bensch R, Çiçek Ö, Abdulkadir A, Marrakchi Y, Böhm A, Deubner J, Jäckel Z, Seiwald K, Dovzhenko A, Tietz O, Dal Bosco C, Walsh S, Saltukoglu D, Tay TL, Prinz M, Palme K, Simons M, Diester I, Brox T, Ronneberger O. U-Net: deep learning for cell counting, detection, and morphometry. Nat. Methods 2018. Available at: https://doi.org/10.1038/s41592-018-0261-2.

15. Lin T-Y, Dollár P, Girshick R, He K, Hariharan B, Belongie S. Feature Pyramid Networks for Object Detection. arXiv [cs.CV] 2016. Available at: http://arxiv.org/abs/1612.03144.

16. He K, Gkioxari G, Dollár P, Girshick R. Mask R-CNN. arXiv [cs.CV] 2017. Available at: http://arxiv.org/abs/1703.06870.

17. Gadkari M, Makiya MA, Legrand F, Stokes K, Brown T, Howe K, Khoury P, Hu Z, Klion A, Franco LM. Transcript- and protein-level analyses of the response of human eosinophils to glucocorticoids. Sci Data 2018;5:180275.

18. Köster J, Rahmann S. Snakemake--a scalable bioinformatics workflow engine. Bioinformatics 2012;28:2520–2522.

19. Fillbrunn A, Dietz C, Pfeuffer J, Rahn R, Landrum GA, Berthold MR. KNIME for reproducible cross-domain analysis of life science data. J. Biotechnol. 2017;261:149–156.

20. Dietz C, Berthold MR. KNIME for Open-Source Bioimage Analysis: A Tutorial. Adv. Anat. Embryol. Cell Biol. 2016;219:179–197.

21. Anon. Supervisely - Web platform for computer vision. Annotation, training and deploy. Available at: https://supervise.ly/. Accessed March 18, 2020.

22. Jung A. imgaug. Github Available at: https://github.com/aleju/imgaug. Accessed March 4, 2020.

23. Anon. Mask_RCNN. Github Available at: https://github.com/matterport/Mask_RCNN. Accessed March 19, 2020.

24. Yakubovskiy P. segmentation_models. Github Available at: https://github.com/qubvel/segmentation_models. Accessed March 19, 2020.

25. Meyer F, Beucher S. Morphological segmentation. Journal of Visual Communication and Image Representation 1990;1:21–46. Available at: http://dx.doi.org/10.1016/1047-3203(90)90014-m.

26. Kandaswamy C, Silva LM, Alexandre LA, Santos JM. High-Content Analysis of Breast Cancer Using Single-Cell Deep Transfer Learning. J. Biomol. Screen. 2016;21:252–259.

27. Raghu M, Zhang C, Kleinberg J, Bengio S. Transfusion: Understanding Transfer Learning for Medical Imaging. arXiv [cs.CV] 2019. Available at: http://arxiv.org/abs/1902.07208.

28. Deng J, Dong W, Socher R, Li L-J, Li K, Fei-Fei L. ImageNet: A Large-Scale Hierarchical Image

Database. In: CVPR09. 2009.

29. Lin T-Y, Maire M, Belongie S, Bourdev L, Girshick R, Hays J, Perona P, Ramanan D, Lawrence Zitnick C, Dollár P. Microsoft COCO: Common Objects in Context. arXiv [cs.CV] 2014. Available at: http://arxiv.org/abs/1405.0312.

30. Anon. Jacobkie. Github Available at: https://github.com/jacobkie/2018DSB. Accessed March 4, 2020.

Figures Legend

Fig. 1) Deep Learning nuclear segmentation pipelines. A) Schematic representation of the computational pipeline used to train deep learning nuclear instance segmentation models and to use these to predict nucleus labels. When present, nuclear labels are pseudocolored to represent different instances. **B)** Image augmentation strategy**.** Random overlapping patches are oversampled from full Fields of View (FOV) of grayscale images and of nuclear labels images. These patches are then augmented by randomly applying a variety of affine, translational, intensity and blurring transformations. The augmented dataset is then used for supervised training of the deep learning nuclear segmentation models. **C)** Schematic of the FPN2-WS architecture. The FPN2-WS architecture uses two deep learning models to predict a normalized nucleus distance transform image, and a blurred nucleus border image. These images are then used as the seed and border for a traditional watershed segmentation algorithm.

Fig. 2) Nuclear segmentation inference performance of the baseline training strategy for the MRCNN and FPN2-WS model architectures. A) The MRCNN and FPN2-WS models were trained on the MCF10A cell line image dataset using random initialization, and applied inference on images of nuclei from the indicated cell types. The images represent pseudocolored nuclear labels for the ground truth (GT) annotations and for the model inference results. Additionally, the GT XOR Inference panels represent pseudocolored full labels for false negatives at an Intersection over Union (IoU) threshold of 0.7. For true positive and false positive objects at the same threshold, only the difference at the pixel level between GT and inference labels is shown in magenta. Scale bar: 20 µm. **B)** Line plot of the F1 Score at increasing Intersection over Union (IoU) thresholds (See Materials and Methods for details) for the MRCNN and FPN2-WS models trained as indicated in A). The 0.7 IoU threshold is indicated by the dashed grey vertical line.

Fig. 3) Transfer learning can improve the inference performance of nuclear segmentation models. A) Dot plot of the F1 Score at the 0.7 IoU threshold for the nuclear

segmentation models trained either as indicated in Fig. 1A) (Randomized), or by initializing the models with weights obtained by pre-training MRCNN on the COCO dataset, or the FPN2-WS on the ImageNet dataset, respectively. **B)** Dot plot of the F1 Score at the 0.7 IoU threshold for the nuclear segmentation models trained either as indicated in Fig. 2A) (MCF10A), or by using a combination of images of nuclei from the other cell types, as indicated on the x-axis.

Fig. 4) Deep learning segmentation models can be efficiently trained with less data, fewer training cycles, and fewer image pre-processing steps. A) Dot plot of the F1 Score at the 0.7 IoU threshold for the nuclear segmentation models trained with images of nuclei from all cell types as indicated in Fig. 3B) (1X), or by reducing the number of training cycles (0.25 - 0.75X). **B)** Dot plot of the F1 Score at the 0.7 IoU threshold for the nuclear segmentation models trained either from all cell types as indicated in Fig. 3B) (1X, 8,750 image patches per cell line), twice as many patches per cell type (2X), or fewer image patches per cell type (0.125X - 0.5X). **C)** Dot plot of the F1 Score at the 0.7 IoU threshold for the nuclear segmentation models trained on 8,750 image patches per cell type that were augmented as indicated in Fig. 3) (Full), not augmented (None), or were augmented with all but one selected class of augmentation operations (Min_Blur - Min_Scaling). **D)** Dot plot and crossbar plot for the mean and 95% confidence interval crossbar for the F1 Score at the 0.7 IoU threshold. Nuclear segmentation models were independently trained with 8,750 image patches for each cell type, for 20 cycles, and as indicated on the x-axis. n = 15 for the MRCNN Full/Random combination, and n = 4 for all other combinations.

Fig. 5) Nuclear segmentation inference performance of the optimized training strategy for the MRCNN and FPN2-WS model architectures, and comparison to the pre-trained Jacobkie model architecture. A) The MRCNN and FPN2-WS models were trained as indicated in Fig. 4D) (None/Pre-initialized for MRCNN, Full/Pre-initialized for FPN2-WS). The Jacobkie model was run using the pre-trained weights from the Kaggle Data Science competition (See Materials and Methods for details). The images represent pseudocolored nuclear labels for the ground truth (GT) annotations and for the models

inference results. Additionally, the GT XOR Inference panels represent pseudocolored full labels for false negatives at an Intersection over Union (IoU) threshold of 0.7. For true positive and false positive objects at the same threshold, only the difference at the pixel level between GT and inference labels is shown in magenta. Scale bar: 20 µm. **B)** Line plot of the F1 Score at increasing IoU thresholds for the MRCNN and FPN2-WS models trained as indicated in A), and for the pre-trained Jacobkie model. The 0.7 IoU threshold is indicated by the dashed grey vertical line.

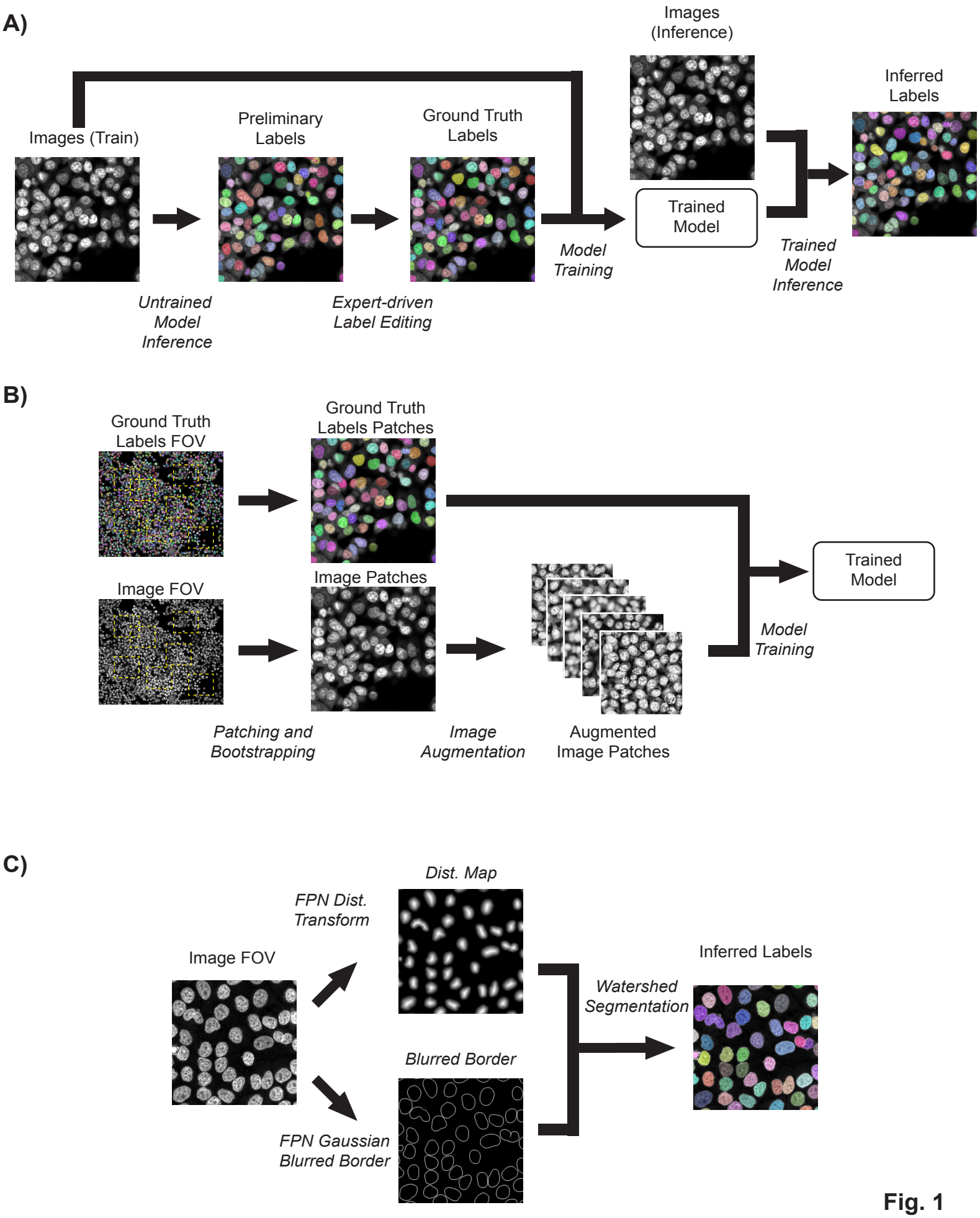
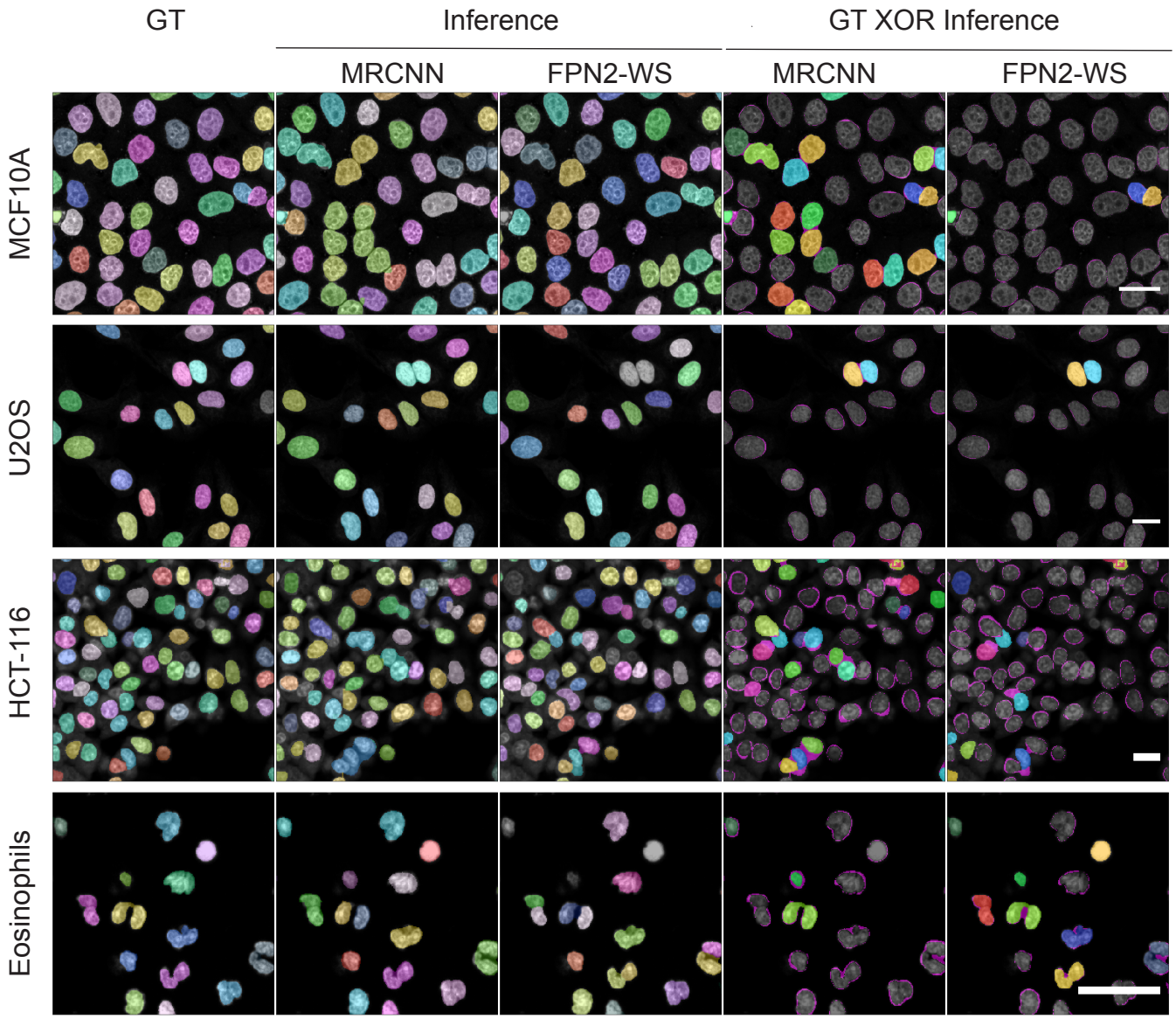

A)
Images (Train)
Preliminary Labels
Ground Truth Labels
Images (Inference)
Inferred Labels
Trained Model
Untrained Model Inference
Expert-driven Label Editing
Model Training
Trained Model Inference
B)
Ground Truth Labels FOV
Ground Truth Labels Patches
Image FOV
Image Patches
Trained Model
Patching and Bootstrapping
Image Augmentation
Augmented Image Patches
Model Training
C)
Dist. Map
FPN Dist. Transform
Image FOV
Inferred Labels
Watershed Segmentation
Blurred Border
FPN Gaussian Blurred Border

Fig. 1

A)


GT
Inference
GT XOR Inference
MRCNN
FPN2-WS
MRCNN
FPN2-WS
MCF10A
U2OS
HCT-116
Eosinophils

B)

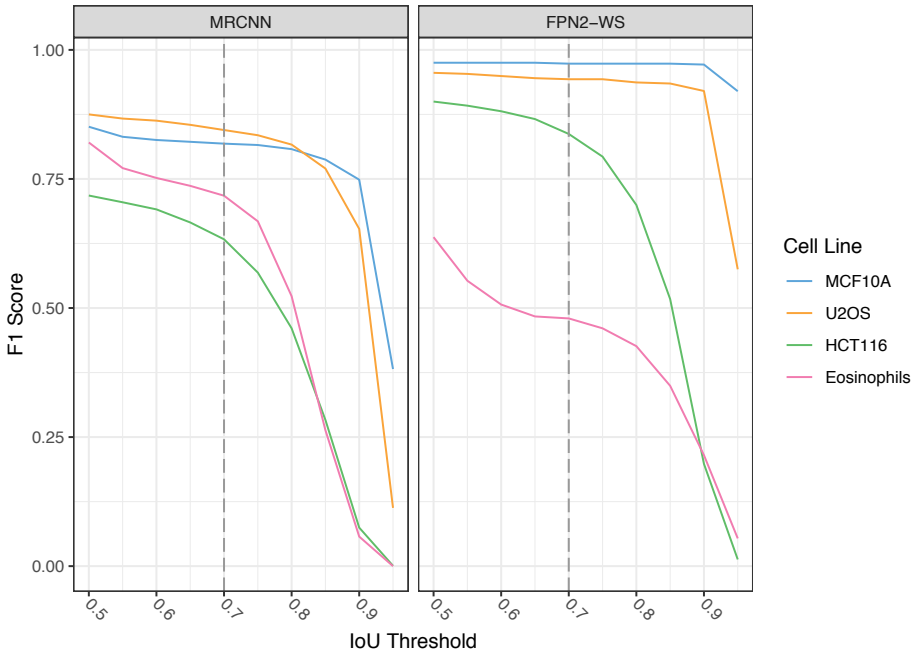
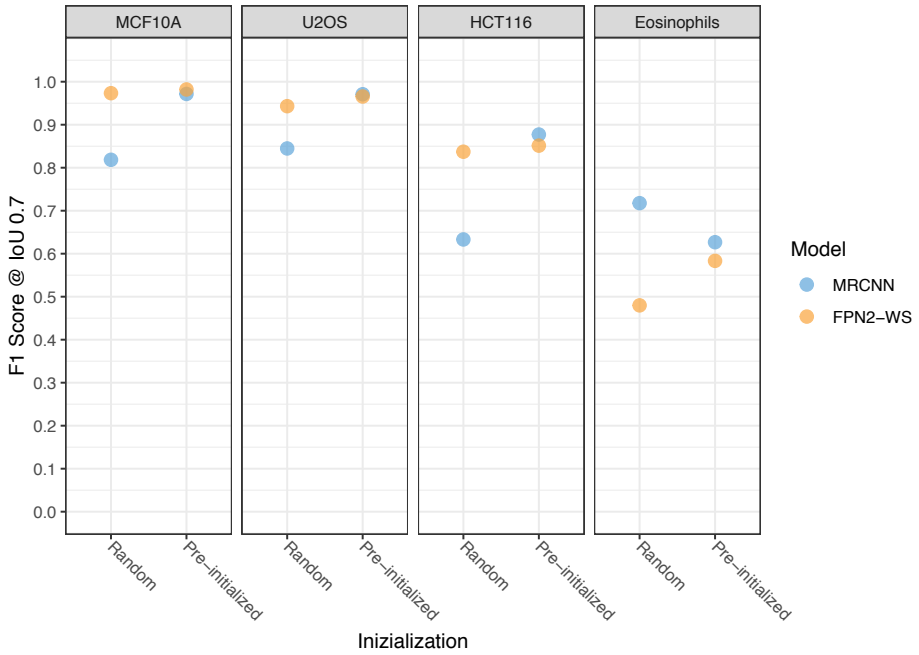

MRCNN
FPN2–WS
F1 Score
1.00
0.75
0.50
0.25
0.00
0.5
0.6
0.7
0.8
0.9
IoU Threshold
Cell Line
MCF10A
U2OS
HCT116
Eosinophils

Fig. 2

A)


MCF10A
U2OS
HCT116
Eosinophils
F1 Score @ IoU 0.7
1.0
0.9
0.8
0.7
0.6
0.5
0.4
0.3
0.2
0.1
0.0
Random
Pre–initialized
Inizialization
Model
MRCNN
FPN2–WS

B)

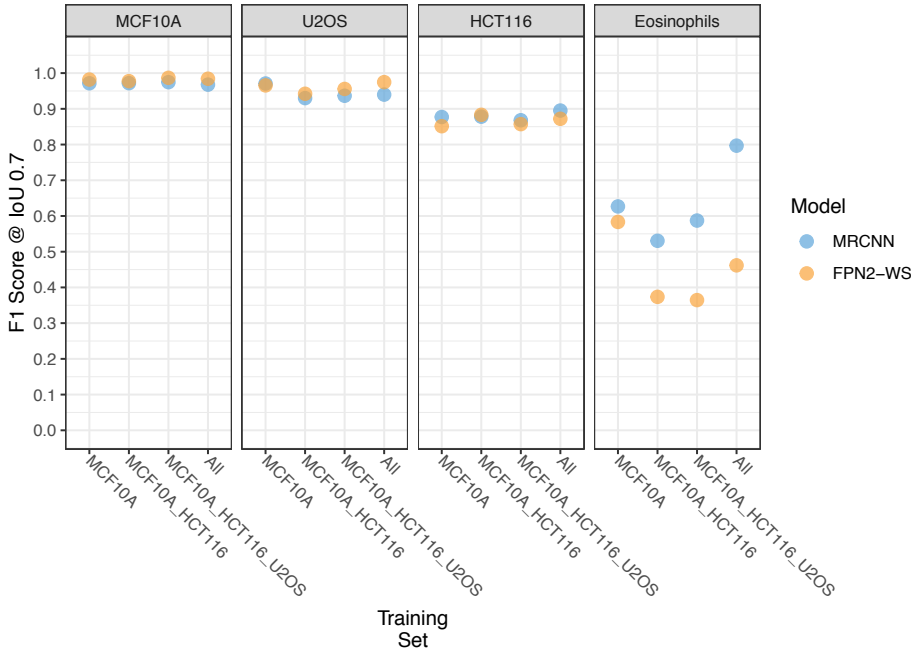

MCF10A
U2OS
HCT116
Eosinophils
F1 Score @ IoU 0.7
1.0
0.9
0.8
0.7
0.6
0.5
0.4
0.3
0.2
0.1
0.0
MCF10A
MCF10A_HCT116
MCF10A_HCT116_U2OS
All
Training Set
Model
MRCNN
FPN2–WS

Fig. 3

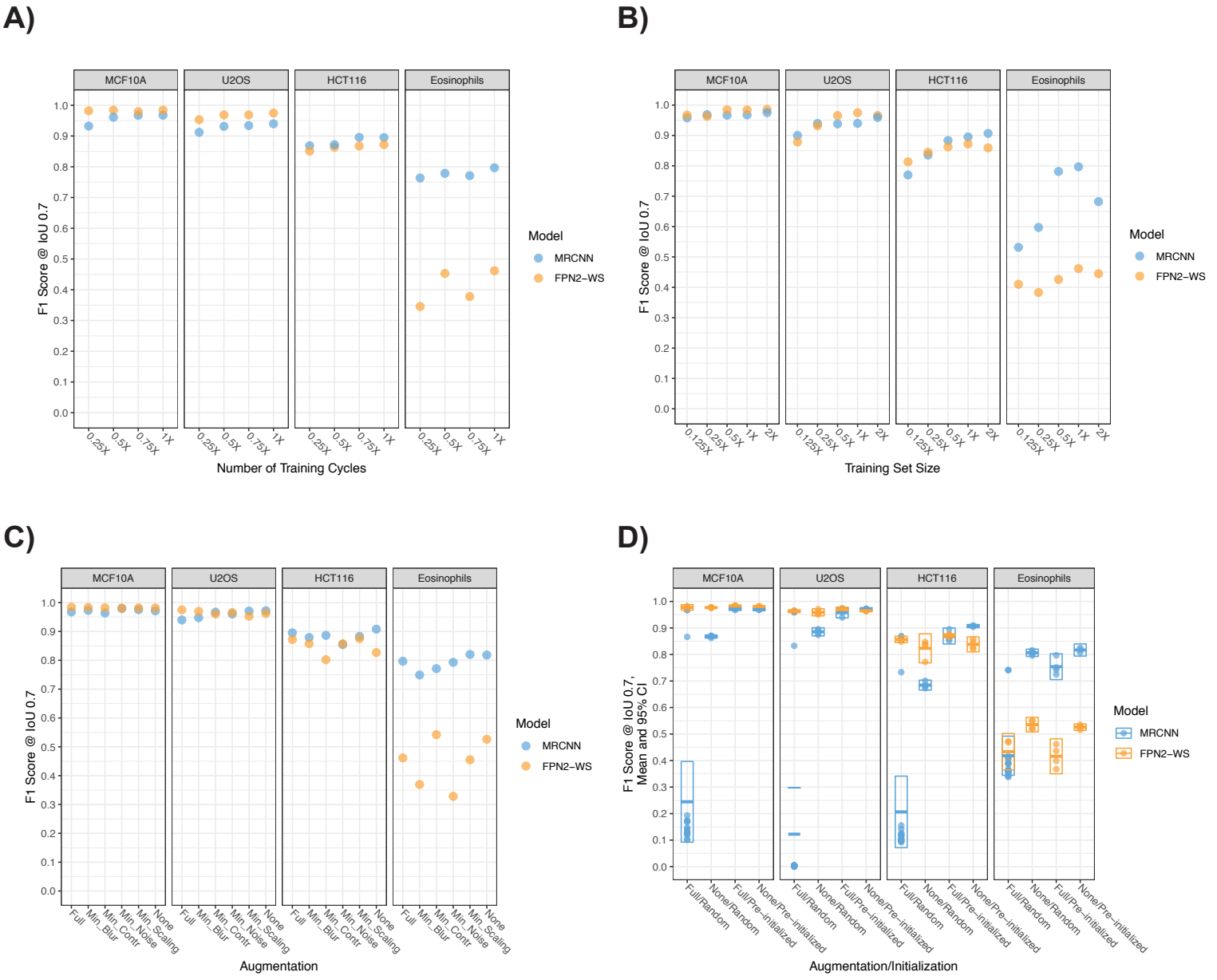
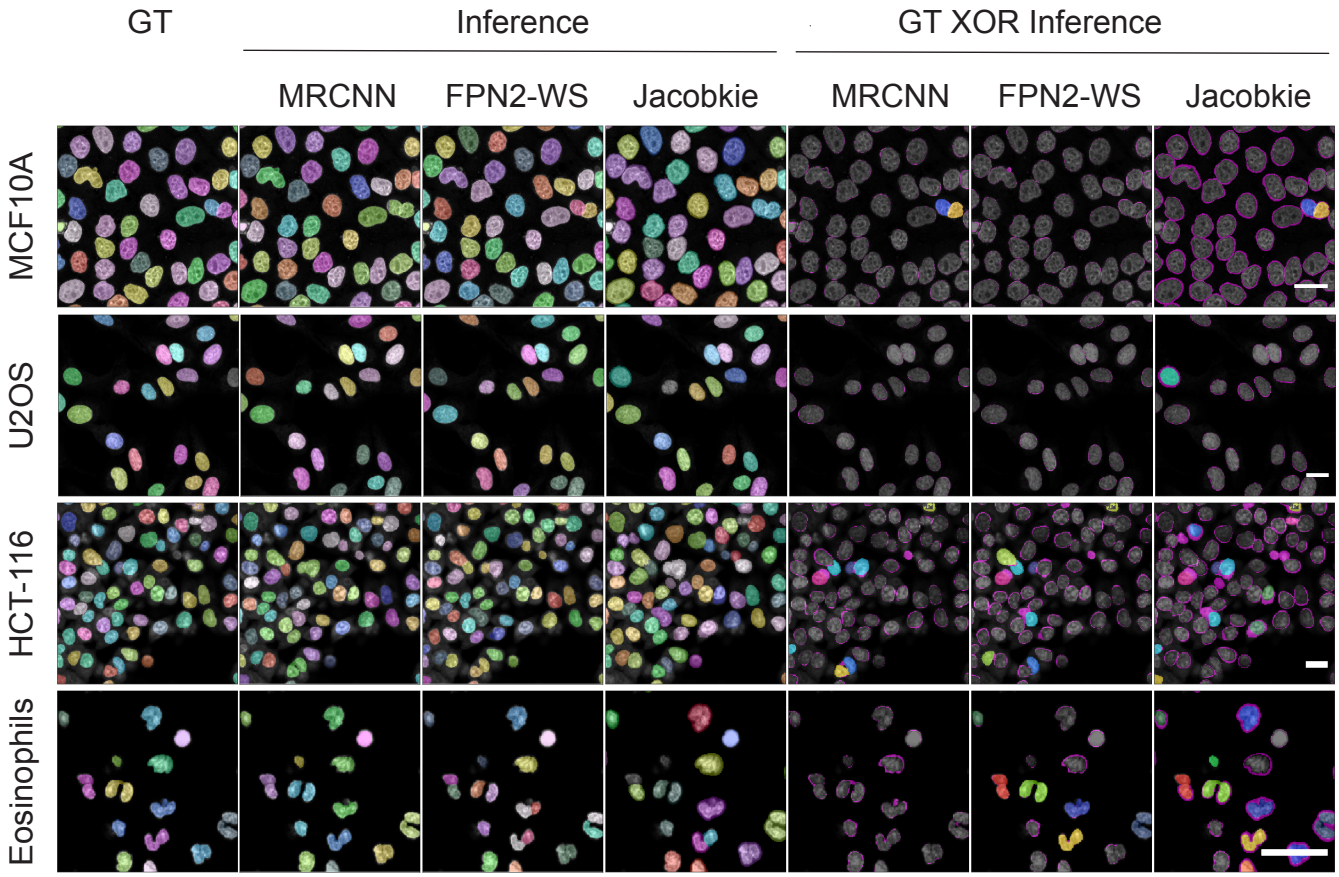

A)
MCF10A
U2OS
HCT116
Eosinophils
F1 Score @ IoU 0.7
Number of Training Cycles
0.25X
0.5X
0.75X
1X
Model
MRCNN
FPN2–WS
B)
Training Set Size
0.125X
0.25X
0.5X
1X
2X
C)
Augmentation
Full
Min_Blur
Min_Contr
Min_Noise
Min_Scaling
None
D)
F1 Score @ IoU 0.7, Mean and 95% CI
Augmentation/Initialization
Full/Random
None/Random
Full/Pre–initialized
None/Pre–initialized

Fig. 4

A)


GT
Inference
GT XOR Inference
MRCNN
FPN2-WS
Jacobkie
MRCNN
FPN2-WS
Jacobkie
MCF10A
U2OS
HCT-116
Eosinophils

B)

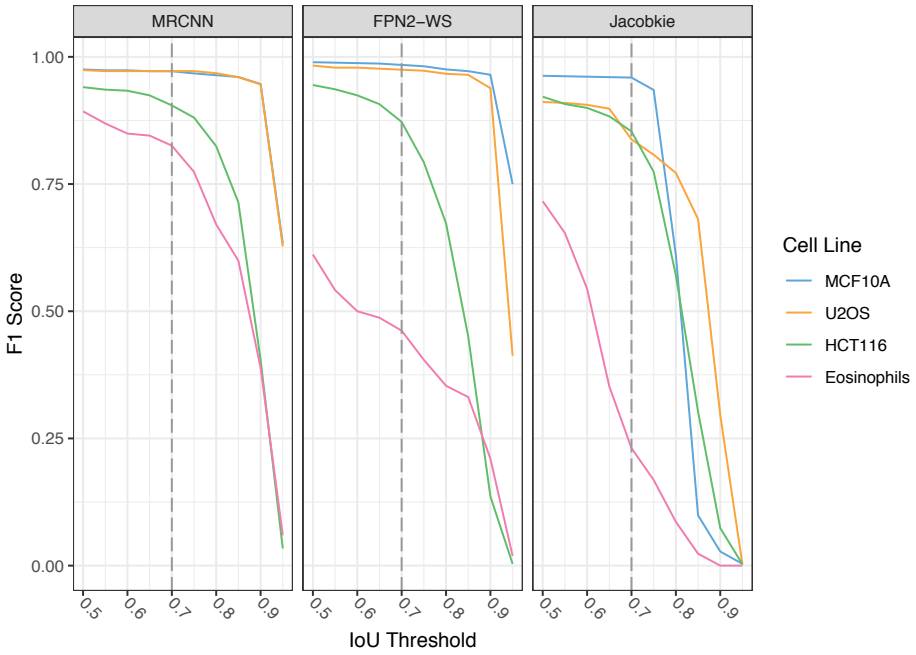

MRCNN
FPN2–WS
Jacobkie
F1 Score
1.00
0.75
0.50
0.25
0.00
0.5
0.6
0.7
0.8
0.9
IoU Threshold
Cell Line
MCF10A
U2OS
HCT116
Eosinophils

Fig. 5