

# Interphase FISH-analysis Processing of Chromosomal Mosaicism on Blood Cell Nuclei Preparations

Alexander Trykin <sup>1</sup>

<sup>1</sup> Lobachevsky State University of Nizhny Novgorod (UNN), 23 Gagarin Ave, Nizhny Novgorod, 603022, Russian Federation

## Abstract

Fluorescence in situ hybridization (FISH) is a molecular cytogenetic technique. It provides a powerful tool for understanding genetic and genomic processes, diagnosing genetic disorders, and studying the structure and function of genes and chromosomes. This paper proposes a method for automatic object segmentation of preparations of blood cell nuclei and a method for detecting chromosomes with the aim of further studying them for chromosomal mosaicism. Based on the data provided by the laboratory of the Institute of Biology and Biomedicine of Lobachevsky University, the SOTA deep learning model YOLOv8-seg was trained. This was made possible by marking up a small portion of the 87 images. Experiment on model training for segmentation showed very good quality metrics: Precision = 0.940, Recall = 0.980, mAP[0.5] = 0.991 and mAP[0.5:0.95] = 0.764. After that, a method for detecting chromosomes was proposed, based on the classical principles of image processing and computer vision, due to the lack of the necessary labelled data.

## Keywords

Fluorescent hybridization, image processing, semantic segmentation, deep neural networks.

## 1. Introduction

Fluorescence in situ hybridization (FISH) technology was developed in the early 1980s by biomedical researchers. This technique was first described by researchers Gall and Pardew in 1969 [1], but it was not until the 1980s that significant advances were made to make FISH a practical and widely used method today [2].

FISH is a molecular cytogenetic technique used to detect and localize specific DNA sequences in cells or tissue samples. For this, fluorescently labelled DNA or RNA probes are used that bind to complementary target sequences in the sample. FISH allows researchers to visualize the spatial organization and distribution of specific genes or chromosomal regions, providing valuable information about gene expression, chromosomal abnormalities, and genomic variation. This method is widely used in various fields, including genetics [3], cancer research [4], prenatal diagnosis [5], and microbial ecology [6]. FISH can be performed on fixed cells or tissue sections, which provides a detailed understanding of the genetic composition and organization of biological samples.

Chromosomal mosaicism refers to the presence of two or more different chromosomal cell populations within an individual. It occurs when there are genetic alterations or abnormalities that affect some but not all cells in the body. This phenomenon can arise during early embryonic development or later in life due to genetic mutations, chromosomal rearrangements, or errors in cell division [7].

In chromosomal mosaicism, some cells possess a normal chromosomal composition, while others exhibit abnormal chromosomal structures or aneuploids (an abnormal number of chromosomes). The degree of mosaicism can vary, ranging from a small percentage of affected cells to a larger proportion.

Mosaicism is often detected through genetic testing methods, such as karyotyping, fluorescence in situ hybridization, or chromosomal microarray analysis. Understanding and diagnosing chromosomal mosaicism is crucial for proper medical management, genetic counselling, and personalized treatment approaches.

---

GraphiCon 2023: 33rd International Conference on Computer Graphics and Vision, September 19-21, 2023

V.A. Trapeznikov Institute of Control Sciences of Russian Academy of Sciences, Moscow, Russia

EMAIL: alexander.trykin@gmail.com (A. Trykin)

ORCID: 0000-0002-5731-9624 (A. Trykin)



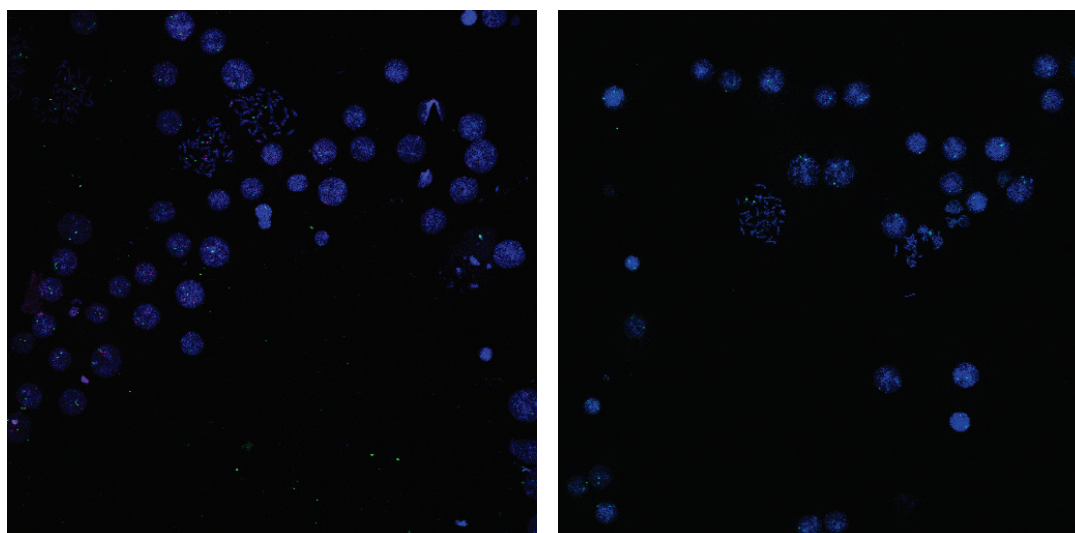
© 2023 Copyright for this paper by its authors.

Use permitted under Creative Commons License Attribution 4.0 International (CC BY 4.0).

Existing methods for segmenting cell nuclei and detecting points (or foci) have made significant progress in automating and expanding the analysis of cellular structures, especially in recent decades with the development of machine and deep learning. For example, The Cell Tracking Challenge, which has become a benchmark in cell segmentation and the development of tracking algorithms [8]. The U-Net architecture (and its modifications) has been one of the most efficient cell segmentation approaches in multiple contrast-enhanced datasets [9, 10]. R-CNN (Region-based Convolutional Neural Network) was also used in this task, but to a lesser extent [11].

## 2. Dataset

The data transmitted to us was collected by the laboratory of the Institute of Biology and Biomedicine of Lobachevsky University. Most images have a resolution of 512x512 pixels, but there are a few images with atypical resolutions of 1024x1024, 1293x1293 and 2873x2873. If we talk about the physical size of a pixel, then it varies from 0.12 to 0.69 nm. This is a mixed dataset of approximately 200 unlabelled images in uncompressed CZI or BMP formats, as well as compressed JPG. The figure 1 shows several examples from the dataset.



**Figure 1:** Several examples from dataset. Blue areas in the image represent cell nuclei, in particular, rounded areas are whole cells, and clusters of small areas are destroyed cells. The red dots represent the locations of the 21-chromosomes and the green dots represent the X-chromosomes.

Brightness/contrast boost applied for better visibility

Notably, the CZI format was developed by ZEISS specifically to meet the imaging requirements of microscopy. It stores not only the images themselves, but also meta-information, such as microscope type and settings, laser or illumination intensities, dimensions and scale bars, used contrasts and objectives, fluorescent dyes and filters, camera and detector parameters such as exposure values, binning, bit depth, pixel dwell time and more. Such information can also be extremely useful in processing images obtained using FISH-analysis. According to the metadata, these 3-channel images are not true RGB, but they can be interpreted as RGB.

Let's talk more about how this data was collected. The study group included 16 subjects with idiopathic infertility referred from medical genetic centres and clinics providing medical services in the field of obstetrics and gynaecology. The age of the subjects varied from 25 to 35 years.

Indirect preparations were prepared from PHA-stimulated peripheral blood lymphocytes cultured for 72 hours, stained with the differential GTG method, followed by analysis. At least 30 metaphase plates of chromosomes of each subject were analysed by the cytogenetic method on a «ZEISS Primo Star» microscope in transmitted light under oil immersion at a total magnification  $\times 1000$ . In cases when aneuploidy on the X-chromosome of individual cells of the sample was detected, the preparation was sent for FISH-analysis. Cytogenetic preparations of peripheral blood lymphocytes and buccal epithelial cells were used for FISH.

Viewing and analysis of preparations was carried out using a confocal fluorescence «ZEISS LSM 800» microscope, where, under the influence of exciting light of a certain spectrum, fluorochrome molecules began to emit light quanta, the wavelength of which depended on the properties of the fluorochrome used. To detect signals from the X-chromosome, the survey was carried out at a wavelength of  $525\pm 30$  nm. For detection of signals from 21-chromosome at a wavelength of  $580\pm 30$  nm.

### 3. Proposed methods

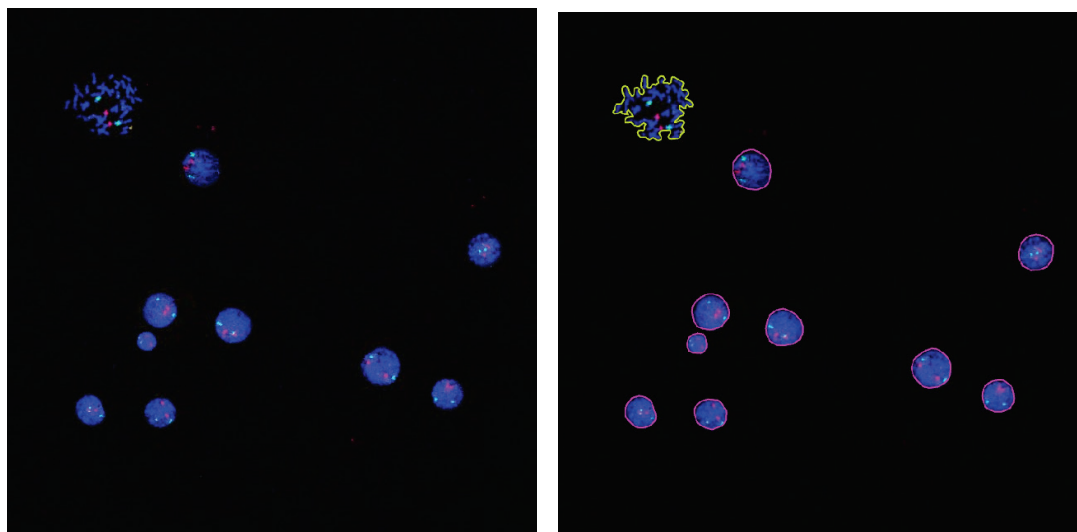
In this paper, a division into 2 tasks is proposed. In the first task, we are trying to segment blood cell nuclei using the YOLOv8 deep learning model. In the second -- to detect chromosomes inside the cell nuclei using the obtained masks from the 1st algorithm.

Also, due to the lack of any labelled data, we are faced with a choice: use unsupervised methods (including classical computer vision methods) or manually label the data in order to later use it to train machine learning models.

Our decision at this stage is to use partial manual labelling for cell nucleus segmentation and not to use labelling for chromosome detection.

#### 3.1. Manual data labelling

As mentioned earlier, we want to label a portion of the images in order to apply the segmentation network. A small part consisting of 87 images was selected. On these images, all cell nuclei were split into 2 groups: healthy (i.e., whole) and destroyed (see example on figure 2). Open-source software was used for this task.



**Figure 2:** An example of manual annotation for segmenting cell nuclei into healthy (red contour) and destroyed (yellow contour)

#### 3.2. Cell nuclei segmentation

Here, we describe which segmentation model we have chosen. Our choice fell on state-of-the-art (SOTA) YOLOv8 algorithm.

YOLO (You Only Look Once) is a popular series of one-stage deep learning models primarily designed for object detection tasks rather than segmentation. The first version of YOLO was presented by Joseph Redmon along with his co-authors Ali Farhadi and Santosh Divwala in 2015 [12]. YOLO has gained popularity due to its excellent trade-off between high speed and accuracy. Since its initial release, YOLO has undergone several iterations and improvements, with subsequent versions being developed by various computer vision communities [13, 14, 15, 16, 17].

Initially, YOLOs are designed for object detection, not segmentation. YOLOv5, although not an official release by the original creators of YOLO, introduced some modifications and improvements to the architecture, including support for segmentation tasks [18]

YOLOv8 is the latest version of the YOLO object detection model introduced by Ultralytics and launched on January 2023, to improve accuracy and efficiency over previous versions [19]. Major updates include a more optimized network architecture (see on figure 5), a revised anchor block design, and a modified loss function to improve accuracy.

YOLOv8 also provides a semantic segmentation model called YOLOv8-Seg model. The backbone is a CSPDarknet53 feature extractor, followed by a C2f module instead of the traditional YOLO neck architecture. The C2f module is followed by two segmentation heads, which learn to predict the semantic segmentation masks for the input image [20].

### 3.3. Chromosome detection

Chromosomes are bright red or green dots on the image, depending on their type. They can look like single dots or double dots due to the spatial position they occupy in the cell.

Therefore, we are pursuing the goal of finding these bright points in the image. We also want to use the cell segmentation masks from the previous algorithm to find chromosomes within cells.

Proposed chromosome detection algorithm:

1. apply unsharp masking separately for red and green channels;
2. potential chromosome candidates are selected:
  - a. image is binarized by the threshold  $T$ ;
  - b. operation of the morphological gradient with a kernel  $3 \times 3$  is applied;
  - c. connected-component labelling is applied to the resulting areas.
3. apply filtering of potential chromosome candidates:
  - a. for each chromosome, the cell nucleus belonging to it is detected, if the chromosome does not lie inside it, then it is excluded from consideration;
  - b. the proximity of the chromosome to the border of the cell is checked, if it lies on it (or is very close in terms of threshold  $C$ ), then the chromosome is excluded from consideration.
4. as a result, cell nuclei with detected chromosomes are obtained.

Now let's discuss the choice of values for the parameters  $T$  and  $C$  introduced above. The  $C$  parameter is set to 1, which means that only the closest points to the boundary will be rejected. The  $T$  parameter is currently set to 80. Both parameters are set for the entire data set and do not depend on a particular image.

## 4. Implementation details

The proposed algorithm is implemented using Python 3.10. For image processing, popular libraries were used, such as OpenCV [21], scikit-image [22], SciPy [23].

YOLOv8s-seg model is taken from Ultralytics with PyTorch backend under the AGPL-3.0 licence [24]. This model with a standard input of  $640 \times 640 \times 3$  has about 11.8 million trainable parameters.

## 5. Experimental results

### 5.1. Evaluation criteria

We will describe the evaluation criteria for the segmentation model. Criteria for evaluating the detection of chromosomes will not be described below due to the lack of necessary markup.

YOLOv8-seg by default uses the following loss function for optimization:

$$L = L_{box} + L_{seg} + L_{obj} + L_{cls}, \quad (1)$$

where  $L_{box}$  uses Complete Intersection over Union (CIoU) [25] and Distributional Focal Loss (DFL) [26],  $L_{seg}$  equivalent to Binary Cross-Entropy (BCE) for image masks,  $L_{cls}$  also equivalent to BCE. For more detailed information, please refer to the relevant citations.

**Intersection over Union (IoU)** is a metric commonly used to evaluate the performance of object detection and segmentation algorithms. It measures the overlap between the predicted bounding box (or segmentation mask)  $B_p$  and the ground truth area  $B_{gt}$ . The formula for IoU is:

$$IoU = \frac{area(B_p \cap B_{gt})}{area(B_p \cup B_{gt})} \quad (2)$$

**True Positive (TP)**, **False Positive (FP)**, **False Negative (FN)**, and **True Negative (TN)** are terms used in binary classification tasks to describe the outcomes of predictions compared to ground truth labels. Some basic concepts used by the metrics:

- True Positive (TP): A correct detection. Detection with  $IoU \geq \text{threshold}$ ;
- False Positive (FP): A wrong detection. Detection with  $IoU < \text{threshold}$ ;
- False Negative (FN): A ground truth not detected;
- True Negative (TN): Does not apply.

**Precision** is the ability of a model to identify only the relevant objects. It is the percentage of correct positive predictions and is given by:

$$Precision = \frac{TP}{TP + FP} = \frac{TP}{\text{all detections}} \quad (3)$$

**Recall** is the ability of a model to find all the relevant cases (all ground truths). It is the percentage of true positive detected among all relevant ground truths and is given by:

$$Recall = \frac{TP}{TP + FN} = \frac{TP}{\text{all ground truths}} \quad (4)$$

**Mean Average Precision (mAP)** is a common evaluation metric used in object detection and instance segmentation tasks. It measures the accuracy and quality of the predicted bounding boxes or segmentation masks. AP (Average Precision) is calculated for each class individually, and then the mAP is the mean of the AP values across all classes.

For our work, we chose mAP[0.5] and mAP[0.5:0.95]. mAP[0.5] refers to the mean Average Precision calculated on IoU threshold equals 0.5. mAP[0.5:0.95] refers to the mean Average Precision calculated over a range of IoU thresholds. The range 0.5:0.95 specifies a set of IoU thresholds. It means that for each class, the AP is calculated by considering the precision and recall values at IoU thresholds ranging from 0.5 to 0.95, with a certain step size equals 0.05. For more details on the calculation of this metric, refer to [27].

## 5.2. Discussion of results

All images were scaled to 640x640x3. 87 images were manually segmented. 60 of them were used for model training and 27 for validation.

Due to the small data set, augmentation was applied. Models from Ultralytics provide an internal mechanism for setting parameters for various augmentations. Also, it has integration with the Albumentations library for fast and flexible modification of images [28]. Albumentations efficiently implements a wide range of image transformation operations optimized for performance. In the current work, only the internal mechanism was used, but not the Albumentations library. Let us describe the types of augmentations used and their parameters:

- image HSV-Hue augmentation (fraction set to 0.015),
- image HSV-Saturation augmentation (fraction set to 0.1),
- image HSV-Value augmentation (fraction set to 0.1),
- image rotation (+/- deg set to 0.25),
- image translation (+/- fraction set to 0.25),
- image scale (+/- gain set to 0.5),

- image shear (+/- deg set to 0.25),
- image perspective (+/- fraction set to 0.001),
- image flip up-down (probability set to 0.25),
- image flip left-right (probability set to 0.25),
- image mosaic (probability set to 1.0).

For YOLOv8, an early stop was used to obtain the best model for the validation loss. The AdamW optimizer was used with default learning rate = 0.01 and momentum = 0.9. We set the threshold of IoU at 0.5. We can report achieved metrics on validation data on table 1. Also check the learning curves on figure 6.

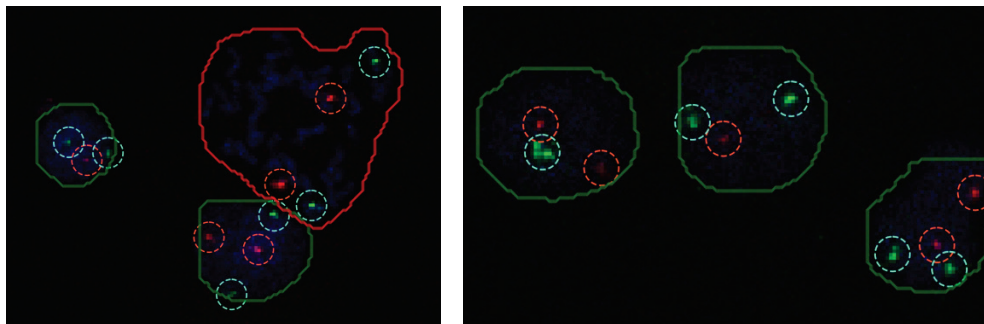
**Table 1:** Precision, Recall, and mAP metric values for the validation dataset

Metric	Value
Precision	0.940
Recall	0.980
mAP[0.5]	0.991
mAP[0.5:0.95]	0.764

Despite the small dataset, very good results were achieved. This can probably be explained by the good generalizing ability of the model and the use of data augmentation.

Let's look at some examples below in figures 3 and 4.

First, let's focus on good examples. On the left image, there are 2 whole cells and one destroyed. Their segmented masks are quite precise, but you can see very little overlap between a whole and a destroyed cell. At the same time, all chromosomes in these cells were found, including the destroyed one. On the right image there are 3 whole cells and also all detected chromosomes.



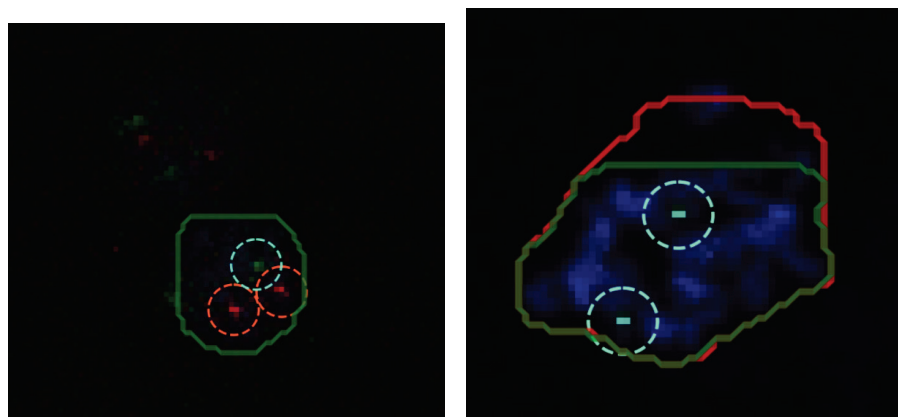
**Figure 3:** Good prediction examples (crop images for better visibility). The green colour of the outline indicates the detected whole cell, and the red colour indicates the destroyed one. The green dotted circle indicates the detected "green" chromosome, and the red circle indicates the "red" one

Now let's switch to bad examples. In the left image, we can assume that the bottom cell was processed acceptably, but the top one was not even segmented, which means the chromosomes were skipped. On the right image is a cell with detected chromosomes, however one and the same cell is marked both as whole and as destroyed.

## 6. Conclusion

In this paper, an algorithm for automatic object segmentation of fluorescent in situ hybridization data was proposed. For this purpose, the SOTA YOLOv8 deep learning model was applied, which is capable of segmenting whole and destroyed cells. For this, a small set of available images was marked up. A non-learning method for detecting chromosomes has also been proposed.

In the course of further work, we want to increase the number of labelled images in order to improve segmentation, propose improvements or revisions to the chromosome detection method, and introduce mosaicism calculation. We also want to consult with experts to get more accurate labelled segmentation masks and chromosomes.



**Figure 4:** Bad prediction examples (crop images for better visibility). The green colour of the outline indicates the detected whole cell, and the red colour indicates the destroyed one. The green dotted circle indicates the detected "green" chromosome, and the red circle indicates the "red" one

## 7. Acknowledgements

This research was supported by The Center of Excellence, Center of Photonics, funded by The Ministry of Science and Higher Education of the Russian Federation, agreement №075-15-2022-293.

## 8. References

- [1] J. G. Gall, M. L. Pardue, Formation and detection of rna-dna hybrid molecules in cytological preparations, *Proceedings of the National Academy of Sciences* 63 (1969) 378–383. URL: <https://doi.org/10.1073%2Fpnas.63.2.378>. doi:10.1073/pnas.63.2.378.
- [2] P. R. Langer-Safer, M. Levine, D. C. Ward, Immunological method for mapping genes on drosophila polytene chromosomes, *Proceedings of the National Academy of Sciences* 79 (1982) 4381–4385. URL: <https://doi.org/10.1073%2Fpnas.79.14.4381>. doi:10.1073/pnas. 79.14.4381.
- [3] A. Wiktor, D. L. V. Dyke, FISH analysis helps identify low-level mosaicism in ullrich-turner syndrome patients, *Genetics in Medicine* 6 (2004) 132–135. URL: <https://doi.org/10.1097%2F01.gim.0000127270.49902.56>. doi:10.1097/01.gim.0000127270.49902.56.
- [4] N. M. Chrzanowska, J. Kowalewski, M. A. Lewandowska, Use of fluorescence in situ hybridization (FISH) in diagnosis and tailored therapies in solid tumors, *Molecules* 25 (2020) 1864. URL: <https://doi.org/10.3390%2Fmolecules25081864>. doi:10.3390/ molecules25081864.
- [5] S. G. Vorsanova, A. D. Kolotii, I. Y. Iourov, V. V. Monakhov, E. A. Kirillova, I. V. Soloviev, Y. B. Yurov, Evidence for high frequency of chromosomal mosaicism in spontaneous abortions revealed by interphase fish analysis, *Journal of Histochemistry & Cytochemistry* 53 (2005) 375–380. doi:10.1369/jhc.4A6424.2005, pMID: 15750024.
- [6] B. Bottari, D. Ercolini, M. Gatti, E. Neviani, Application of FISH technology for microbiological analysis: current state and prospects, *Applied Microbiology and Biotechnology* 73 (2006) 485–494. URL: <https://doi.org/10.1007%2Fs00253-006-0615-z>. doi:10.1007/s00253-006-0615-z.
- [7] B. Levy, E. R. Hoffmann, R. C. McCoy, F. R. Grati, Chromosomal mosaicism: Origins and clinical implications in preimplantation and prenatal diagnosis, *Prenatal Diagnosis* 41 (2021) 631–641. URL: <https://doi.org/10.1002%2Fpd.5931>. doi:10.1002/pd.5931.
- [8] M. Maška, V. Ulman, P. Delgado-Rodriguez, E. G. de Mariscal, T. Necasová, F. A. G. Peña, ˇ T. I. Ren, E. M. Meyerowitz, T. Scherr, K. Löffler, R. Mikut, T. Guo, Y. Wang, J. P. Allebach, R. Bao, N. M. Al-Shakarji, G. Rahmon, I. E. Toubal, K. Palaniappan, F. Lux, P. Matula, K. Sugawara, K. E. G. Magnusson, L. Aho, A. R. Cohen, A. Arbelle, T. Ben-Haim, T. R. Raviv, F. Isensee, P. F. Jäger, K. H. Maier-Hein, Y. Zhu, C. Ederra, A. Urbiola, E. Meijering, A. Cunha, A. Muñoz-Barrutia, M. Kozubek, C. O. de Solórzano, The cell tracking challenge: 10 years of objective benchmarking, *Nature Methods* 20 (2023) 1010–1020. URL: <https://doi.org/10.1038/s41592-023-01879-y>. doi:10.1038/s41592-023-01879-y.

- [9] O. Ronneberger, P. Fischer, T. Brox, U-net: Convolutional networks for biomedical image segmentation, in: *Lecture Notes in Computer Science*, Springer International Publishing, 2015, pp. 234–241. URL: [https://doi.org/10.1007/978-3-319-24574-4\\_28](https://doi.org/10.1007/978-3-319-24574-4_28). doi:10.1007/978-3-319-24574-4\_28.
- [10] F. Long, Microscopy cell nuclei segmentation with enhanced u-net, *BMC Bioinformatics* 21 (2020). URL: <https://doi.org/10.1186/s12859-019-3332-1>. doi:10.1186/s12859-019-3332-1.
- [11] R. Girshick, J. Donahue, T. Darrell, J. Malik, Rich feature hierarchies for accurate object detection and semantic segmentation, in: *2014 IEEE Conference on Computer Vision and Pattern Recognition*, IEEE, 2014. URL: <https://doi.org/10.1109/cvpr.2014.81>. doi:10.1109/cvpr.2014.81.
- [12] J. Redmon, S. Divvala, R. Girshick, A. Farhadi, You only look once: Unified, real-time object detection, in: *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, IEEE, 2016. URL: <https://doi.org/10.1109%2Fcvpr.2016.91>. doi:10.1109/cvpr.2016.91.
- [13] J. Redmon, A. Farhadi, YOLO9000: Better, faster, stronger, in: *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, IEEE, 2017. URL: <https://doi.org/10.1109%2Fcvpr.2017.690>. doi:10.1109/cvpr.2017.690.
- [14] A. Bochkovskiy, C.-Y. Wang, H.-Y. M. Liao, Yolov4: Optimal speed and accuracy of object detection, 2020. URL: <https://arxiv.org/abs/2004.10934>. doi:10.48550/ARXIV.2004.10934.
- [15] Z. Ge, S. Liu, F. Wang, Z. Li, J. Sun, Yolox: Exceeding yolo series in 2021, 2021. URL: <https://arxiv.org/abs/2107.08430>. doi:10.48550/ARXIV.2107.08430.
- [16] C.-Y. Wang, A. Bochkovskiy, H.-Y. M. Liao, Yolov7: Trainable bag-of-freebies sets new state-of-the-art for real-time object detectors, 2022. URL: <https://arxiv.org/abs/2207.02696>. doi:10.48550/ARXIV.2207.02696.
- [17] S. Xu, X. Wang, W. Lv, Q. Chang, C. Cui, K. Deng, G. Wang, Q. Dang, S. Wei, Y. Du, B. Lai, Pp-yoloe: An evolved version of yolo, 2022. URL: <https://arxiv.org/abs/2203.16250>. doi:10.48550/ARXIV.2203.16250.
- [18] G. Jocher, YOLOv5 by Ultralytics, 2020. URL: <https://github.com/ultralytics/yolov5>. doi:10.5281/zenodo.3908559.
- [19] G. Jocher, A. Chaurasia, J. Qiu, Ultralytics yolov8, 2023. URL: <https://github.com/ultralytics/ultralytics>.
- [20] J. Terven, D. Cordova-Esparza, A comprehensive review of yolo: From yolov1 and beyond, 2023. URL: <https://arxiv.org/abs/2304.00501>. doi:10.48550/ARXIV.2304.00501.
- [21] G. Bradski, The OpenCV Library, *Dr. Dobb's Journal of Software Tools* (2000).
- [22] S. Van der Walt, J. L. Schönberger, J. Nunez-Iglesias, F. Boulogne, J. D. Warner, N. Yager, E. Gouillart, T. Yu, scikit-image: image processing in python, *PeerJ* 2 (2014) e453.
- [23] P. Virtanen, R. Gommers, T. E. Oliphant, M. Haberland, T. Reddy, D. Cournapeau, E. Burovski, P. Peterson, W. Weckesser, J. Bright, S. J. van der Walt, M. Brett, J. Wilson, K. J. Millman, N. Mayorov, A. R. J. Nelson, E. Jones, R. Kern, E. Larson, C. J. Carey, I. Polat, Y. Feng, E. W. Moore, J. VanderPlas, D. Laxalde, J. Perktold, R. Cimrman, I. Henriksen, E. A. Quintero, C. R. Harris, A. M. Archibald, A. H. Ribeiro, F. Pedregosa, P. van Mulbregt, SciPy 1.0 Contributors, SciPy 1.0: Fundamental Algorithms for Scientific Computing in Python, *Nature Methods* 17 (2020) 261–272. doi:10.1038/s41592-019-0686-2.
- [24] G. Jocher, A. Chaurasia, J. Qiu, YOLO by Ultralytics, 2023. URL: <https://github.com/ultralytics/ultralytics>.
- [25] Z. Zheng, P. Wang, W. Liu, J. Li, R. Ye, D. Ren, Distance-iou loss: Faster and better learning for bounding box regression, 2019. URL: <https://arxiv.org/abs/1911.08287>. doi:10.48550/ARXIV.1911.08287.
- [26] X. Li, W. Wang, L. Wu, S. Chen, X. Hu, J. Li, J. Tang, J. Yang, Generalized focal loss: Learning qualified and distributed bounding boxes for dense object detection, 2020. URL: <https://arxiv.org/abs/2006.04388>. doi:10.48550/ARXIV.2006.04388.
- [27] C. Feichtenhofer, A. Pinz, A. Zisserman, Detect to track and track to detect, 2017. URL: <https://arxiv.org/abs/1710.03958>. doi:10.48550/ARXIV.1710.03958.
- [28] A. Buslaev, V. I. Iglovikov, E. Khvedchenya, A. Parinov, M. Druzhinin, A. A. Kalinin, Albumentations: Fast and flexible image augmentations, *Information* 11 (2020). URL: <https://www.mdpi.com/2078-2489/11/2/125>. doi:10.3390/info11020125.





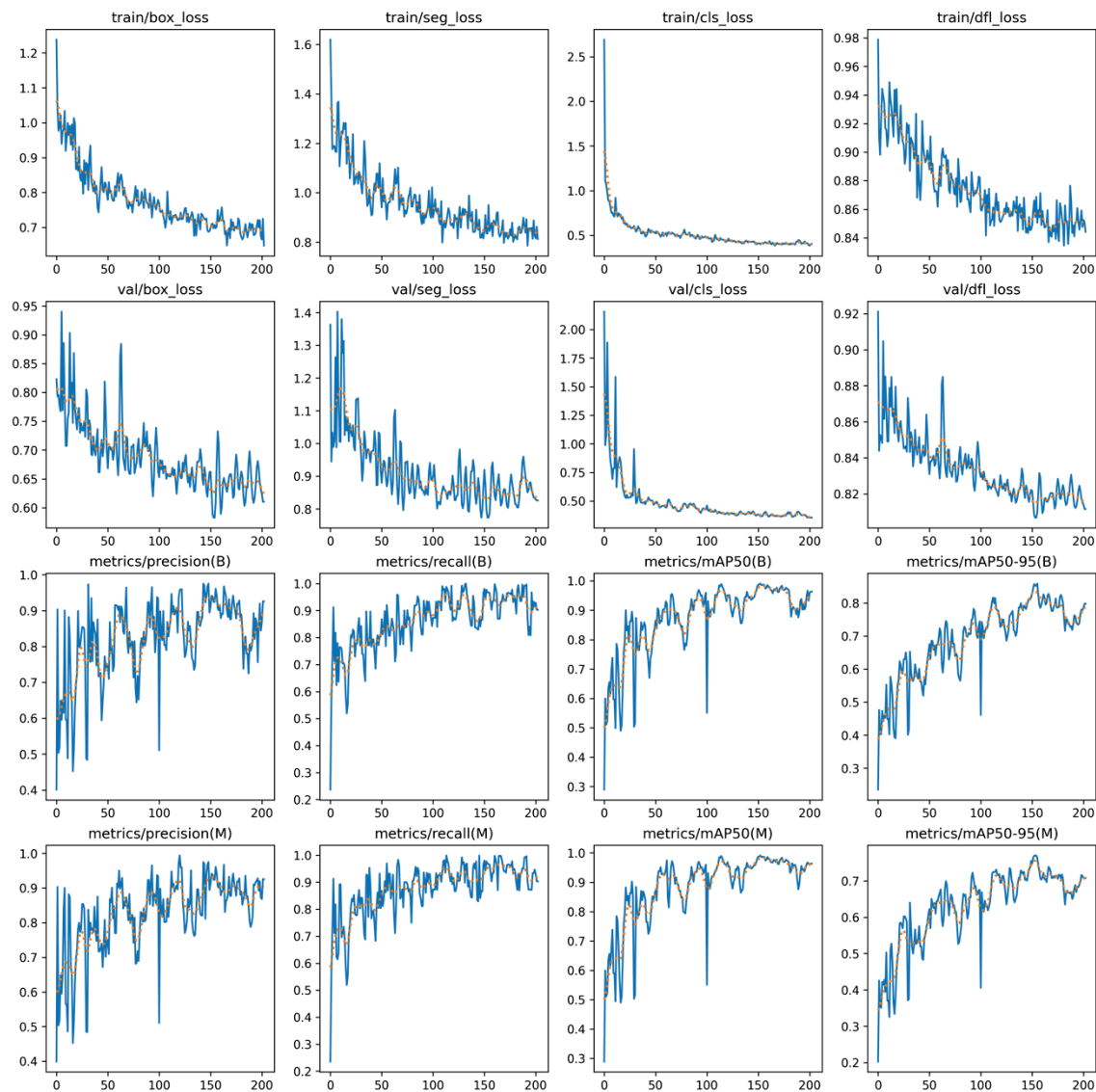


Figure 6: Model learning curves (loss and metrics)