УДК 378:004
DOI: 10.25686/978-5-8158-2474-4-2025-844-851

Локализация клеток с использованием моделей на основе YOLO в анализе жизнеспособности живых-мертвых клеток

Александр Трыкин[1,2], Михаил Кривоносов[1,2], Елизавета Калинина[3]
[1] Исследовательский центр искусственного интеллекта, Институт информационных технологий, математики и механики, ННГУ, Нижний Новгород, Россия
[2] Лаборатория системной медицины старения, НИИ биологии старения, ННГУ, Нижний Новгород, Россия
[3] Институт биологии и биомедицины, ННГУ, Нижний Новгород, Россия

Аннотация. Анализ жизнеспособности клеток критически важен для оценки клеточного здоровья особенно в таких областях, как скрининг лекарственных препаратов, исследования рака и тканевая инженерия. В данной работе мы рассматриваем задачу анализа жизнеспособности живых и мертвых клеток через призму обнаружения объектов, используя модели на основе YOLO для точной локализации и классификации живых и мертвых клеток на микроскопических изображениях. Наш вклад заключается в следующем: (1) представляем полностью аннотированный набор данных для анализа жизнеспособности живых и мертвых клеток в качестве задачи обнаружения объектов; (2) проводим эксперименты с относительно небольшой обучающей выборкой (2-10 % из 252 изображений) и демонстрируем способность нейронной модели к обучению и обобщению, достигая при этом довольно высоких показателей качества; (3) обучаем новейшие модели YOLO (с версии 8 по версию 12) и достигаем высоких и сопоставимых показателей качества для каждой модели; лучшие из представленных моделей достигают точности до 97 % для синих клеток и 96.5 % для красных клеток с достаточным количеством полноты.

Ключевые слова: обработка изображений, флуоресцентная микроскопия, обнаружение объектов, глубокое обучение.

Cell Localization with YOLO-based Models in Live-Dead Cell Viability Assay

Alexander Trykin[1,2], Mikhail Krivonosov[1,2], Elizaveta Kalinina[3]
[1] Artificial Intelligence Research Center, Institute of Information Technologies, Mathematics and Mechanics, Lobachevsky State University, Nizhny Novgorod, Russia
[2] Laboratory of Systems Medicine of Ageing, Research Institute for Biology of Aging, Lobachevsky State University, Nizhny Novgorod, Russia
[3] Institute of Biology and Biomedicine, Lobachevsky State University, Nizhny Novgorod, Russia

Abstract. Cell viability assays are critical for assessing cellular health, particularly in applications such as drug screening, cancer research, and tissue engineering. In this work, we address the challenge of live-dead cell viability analysis through the lens of object detection, utilizing YOLO-based models to accurately localize and classify live and dead cells in microscopy images. Our contributions are as follows: (1) we present a fully annotated dataset for live-dead cell viability analysis as an object detection task; (2) we conduct experiments with a relatively small training sample size (2-10% of 252 images) and demonstrate the ability of the neural model to learn and generalize, while achieving fairly high quality metrics; (3) we train the latest YOLO models (from v8 to v12) and achieve high and comparable quality scores for each model, the best of presented models achieve up to 97% precision for blue cells and 96.5% for red cells with sufficient recall numbers.

Keywords*:* image processing, fluorescence microscopy, object detection, deep learning

Introduction

Live-dead cell viability assays are laboratory methods used to determine the proportion of live and dead cells in a population. These assays are commonly used in cell biology, drug discovery, and biomedical research to assess cell viability, cytotoxicity, and overall cell health.

The principle behind live-dead cell viability assays is to distinguish between live cells with intact membranes (indicating cell viability) and dead cells with compromised membranes (indicating cell death). Various techniques and dyes are utilized to differentiate between these two populations on the basis of their membrane integrity.

In recent years, convolutional neural networks (CNNs) have revolutionized the field of computer vision, and their application in biomedical image analysis has shown immense promise.

The paper [1] proposes a modification of YOLOv5 with FPN (Feature Pyramid Network) to handle cells of different sizes in the images. Their dataset consisted of 283 images, which were reasoned to 600 using techniques such as rotation, scale and flip. They were able to achieve an AP (Average Precision) of 0.8 and a processing time of 43.9 ms per image (~ 144 FPS) on a public dataset.

The work [2] proposes the use of the E-U-Net segmentation network in the task of "live-dead" analysis on unlabelled cells using phase imaging with computational specificity (PICS). They developed a U-Net-like neural network with an EfficientNet encoder using transfer learning on live adherent HeLa and CHO cell cultures. They obtained classification of the viable state of a single cell with an accuracy of 95%.

The paper [3] proposes a YOLOv5-based approach for automatic cell recognition and counting in a case study of laboratory cell detection using images from a CytoSMART Exact FL microscope. The authors used a dataset of 21 expert-labelled images, as well as an additional Sperm DetectionV dataset of 1024 images for transfer learning. The proposed method showed significantly better performance compared to the U-Net model. YOLOv5 had accuracy, precision, recall, and F1 scores of 92%, 84%, 91%, and 87%, respectively.

In paper [4], a novel two-stage DL-based detection method (cGAN-YOLO) is developed to further improve the cell counting efficiency by combining a DL-based transmitted-light images and a DL-based cell detection model. The dataset of transmitted-light images and the fluorescent images consisted of 4000 512 × 512 images of human umbilical vein endothelial cells (HUVEC) and MadinDarby canine kidney cells (MDCK). The recognition accuracy (RA) results were obtained as follows: 96.67% for HUVEC and 92.58% for MDCK.

This study explores the application of YOLO-based models for cell localization in live-dead viability assays. By leveraging the robustness of YOLO algorithms, we aim to provide a method that is not only fast and accurate, but also scalable across different assay conditions and imaging platforms.

Materials

1. Proposed Dataset

The data transmitted to us were collected by the laboratory of the Institute of Biology and Biomedicine of Lobachevsky University. There are a total of 180 pairs of images in PNG format and with a resolution of 2560 × 1920: 180 with blue fluorescent emission and 180 with red fluorescent emission. These pairs belong to different experiments, thereby forming series (sequences). An example of data is shown in fig. 1.

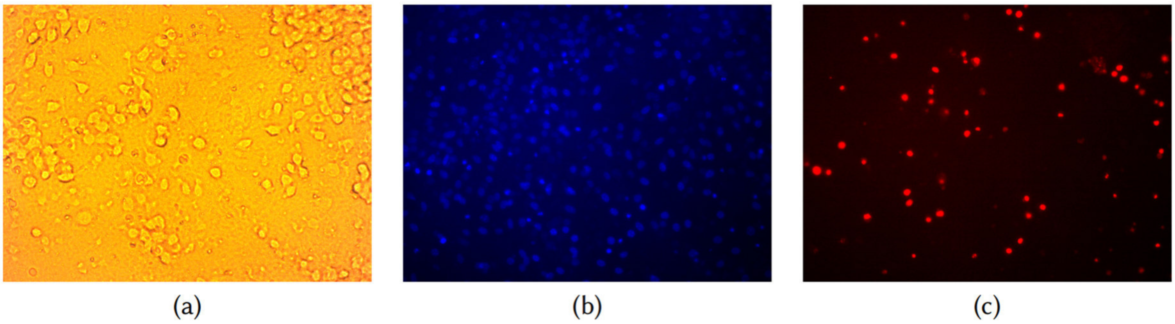


Figure 1. Example images from collected dataset. (a) brightfield/phase-contrast: cell morphology and density are visible without fluorescence, (b) blue fluorescence: dye penetrates nuclei, (c) red fluorescence: indicator of dead cells, cannot cross intact plasma membranes, so it only penetrates cells with compromised membranes

At the beginning of the work, we had only a labelled part of the entire dataset: 34 with a blue emission and 34 with a red emission. We used the Roboflow [5] to perform collaborative management of the dataset, and it's labelling. We have made our dataset public: https://universe. roboflow.com/unn/cytotoxicity-hypoxia-neuroadapt-od (during the work there were several versions of the dataset, v8 is the current one, on which the experiments were conducted).

We prepared the dataset by annotating cells with a bounding box in YOLOv8 Annotation Format. This involved converting the bounding box coordinates to (x, y, w, h) format, where (x, y) represents the coordinates of the centre of the bounding box and (w, h) represents the width and height of the bounding box. The coordinates and dimensions of the box were specified in a normalized form, depending on the height and width of the image. The class label for each bounding box was also encoded as an integer. Quality checks were performed to ensure the accuracy and consistency of the annotations.

After annotation, we randomly split the dataset into training (70%), validation (20%) and test (10%) sets. In any case, researchers are free to use any other split according to their needs.

It is important to note that cell labelling turned out to be quite a complex task, since in some cases even a person cannot always determine with certainty whether an area is a cell or a background. Such situations arise due to the complex structure of cells, low contrast or high noise in the images, which complicates the verification process. As a result, errors in the dataset labelling are possible due to subjective interpretation of complex areas of the image. Despite this, the creation of such a dataset remains an important step for training and evaluating models working with cell image analysis.

2. Preliminary Data Analysis

The cells in the presented images are round and differ in size. Cells are usually discrete but may "clump" together with varying degrees of overlap, which can be a problem in cell detection. The cells themselves have different luminosities, from brightly fluorescent with a fairly clear border to weakly fluorescent with a vague border that almost merges with the background.

In different images, in the same image, but in different parts of the frame, background heterogeneity may be observed. This is clearly visible in example 2 (a), where the background inside the red rectangle is taller (lighter) than the background inside the yellow rectangle (darker).

And in examples 2 (b, c) one can see the possible variation of the background in different images from almost imperceptible to the level of strong noise. This may be due to the peculiarities of the scanning microscope (and/or its settings), the hypothetical spreading of the fluorescent compound from the cells.

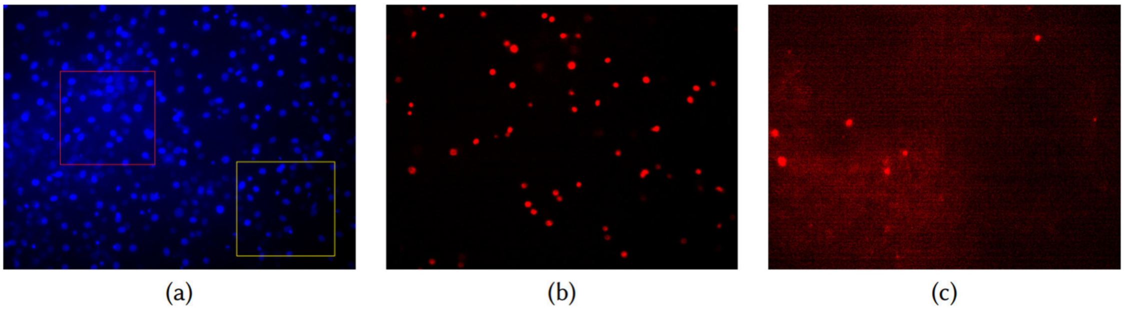


Figure 2. Images showing variations in conditions: in (a) the background inside the red rectangle is lighter than the background inside the yellow rectangle, in (b, c) shows the variability of the background

Methods

1. YOLO's

YOLO (You Only Look Once), a renowned family of single-stage deep learning models, specializes in object detection. Introduced in 2015, YOLO has become widely adopted for its impressive balance of speed and accuracy [6]. Over time, YOLO has evolved through numerous iterations and enhancements, with subsequent versions being continually refined by the computer vision community [7, 8].

YOLOv8 [9]: YOLOv8, released in January 2023 by Ultralytics, introduced several innovations to improve flexibility and accuracy. This version utilized a "C2f module," a variant of the CSPNet module, which optimized feature extraction and model efficiency. It adopted an anchor-free design for bounding box predictions and integrated decoupled heads, independently addressing objectness, classification, and bounding box regression. YOLOv8 also employed CIoU and Distributional Focal Loss (DFL) for bounding box calculations, enhancing localization accuracy specifically for smaller objects like cells.

YOLOv9 [10]: YOLOv9 incorporated advanced methods such as Generalized Efficient Layer Aggregation Network (GELAN) and Programmable Gradient Information (PGI) to further reduce data loss during feature extraction and transformation. The PGI method preserved complete input information for robust gradient computations, ensuring more reliable weight updates and superior convergence.

YOLOv10 [11]: Released in mid-2024, YOLOv10 introduced improvements in computational efficiency and ensured higher accuracy for dense detection scenarios. It utilized multiscale anchor-free detection combined with advanced distributional loss functions like DFL. YOLOv10 addressed issues in detecting rare or clustered objects by modifying its confidence scoring approach and introducing enhanced training for small object instances, which is critical in detecting both live and dead cells with significant morphological overlaps. However, when benchmarked, the accuracy of YOLOv10 was slightly lower than YOLOv9s on datasets involving rare objects, highlighting the importance of contextual model training.

YOLOv11 [12]: YOLOv11, one of the newest iteration of the YOLO series, boasts a streamlined design incorporating C3K2 blocks, SPFF (Spatial Pyramid Pooling Fast), and cutting-edge attention mechanisms such as C2PSA. This new architecture aims to bolster the detection of small objects and elevate overall accuracy, all while preserving the real-time inference speed that has become a hallmark of YOLO.

YOLOv12 [13]: YOLOv12 represents the current pinnacle of YOLO development as of late 2024. It integrates innovations such as self-supervised pretraining, transformer blocks, and enhanced attention mechanisms for improved feature representation. Enhanced quantization-aware training allowed YOLOv12 to operate efficiently even on edge devices, facilitating seamless integration into laboratory equipment for real-time analysis.

1.1 Implementation Details

We used models implemented in Ultralytics [14] with PyTorch backend and AGPL-3.0 licence. In the experiments with the YOLO architectures, it is important to highlight that fine-tuning was performed using the pretrained models YOLOv8-N, YOLOv9-T, YOLOv10-N, YOLOv11-N, and YOLOv12-N. Here, "N" represents the nano version (except "T" which indicates the tiny version), aiming to achieve a balance between speed and precision. Important characteristics of the models, such as the number of weights and operating speed, are shown in table 1.

Table 1. **Summary table of model characteristics**

Model	Weights (M)	FLOPs (G)	CPU Speed (FPS) ONNX	GPU Speed (FPS) TensorRT
YOLOv8-N	3.01	8.2	9.13	261.89
YOLOv9-T	2	7.8	7.8	133.66
YOLOv10-N	2.7	8.4	9.83	141.09
YOLOv11-N	2.59	6.4	9.98	221.77
YOLOv12-N	2.57	6.5	6.22	84.78

The input data shape for each model is 3 × 1280 × 1280 (CHW order), so the images are simply resized to this resolution. The batch size is set to 4. AdamW optimizer was used with an initial learning rate of 1.667 × 10−3 and a momentum of 0.9. Overall, AdamW is an improved version of the Adam optimizer that handles weight regularization more efficiently and, as a result, improves the quality and generalization ability of machine learning models [17]. We also saved the weights of the best performing version of the model for use in testing.

All the algorithms were trained using an NVIDIA RTX 3080 Ti (12 GB) with Python 3.11, PyTorch 2.7 and CUDA 11.8. CPU speeds measured with ONNX export on AMD Ryzen 7 5800X3D, GPU speeds measured with TensorRT export on NVIDIA RTX 3080 Ti. YOLOv12-N was used without FlashAttention [15, 16], which may negatively affect the performance.

1.2 Data Augmentation

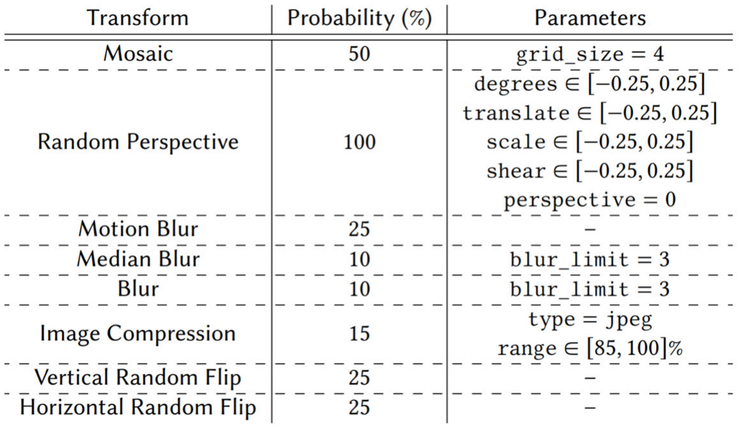
Data augmentation is a vital step in enhancing the robustness and generalizability of deep learning models, particularly in complex tasks [18]. By artificially expanding the dataset through the application of various transformations, data augmentation ensures better model performance by exposing the model to diverse representations of the data.

In the context of YOLO training, several augmentation techniques can be applied to simulate varying conditions that might occur in real-world microscopy images. These techniques are particularly useful for addressing challenges like variability in cell morphology, differences in staining intensities, and diverse imaging conditions.

Ultralytics uses its own implementations of some augmentation techniques, and also uses the well-known Albumentations [19]. Unfortunately, the set of augmentations is already defined and cannot be changed directly. To overcome this, we use a monkey patch and define our own transformations.

The list of used augmentations, as well as their settings, is presented in the table 2

Table 2. **Table of used augmentations, their probabilities and parameters**



Transform	Probability (%)	Parameters
Mosaic	50	grid_size = 4
Random Perspective	100	degrees ∈ $[-0.25, 0.25]$ translate ∈ $[-0.25, 0.25]$ scale ∈ $[-0.25, 0.25]$ shear ∈ $[-0.25, 0.25]$ perspective = 0
Motion Blur	25	–
Median Blur	10	blur_limit = 3
Blur	10	blur_limit = 3
Image Compression	15	type = jpeg range ∈ $[85, 100]$%
Vertical Random Flip	25	–
Horizontal Random Flip	25	–

2. *Evaluation Metrics*

Typically, quality metrics are based on correct and incorrect detection. A common method for this is the use of the intersection over union (IoU). According to [20] the IoU is a measure based on Jaccards index which measures the area of overlap between the predicted bounding box Bp and the true bounding box Bgt divided by the area of union between them:

$$IoU = \frac{area(Bp \ \cap \ Bgt)}{area(Bp \ \cup \ Bgt)}. \tag{1}$$

The value of IoU is then compared with a defined threshold t in order to establish when a detection is correct or incorrect. If IoU ≥ t, then the detection is considered correct, otherwise the detection is considered incorrect.

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

$$Precision = \frac{TP}{TP \ + \ FP} = \frac{TP}{all\ detections}. \tag{2}$$

Recall (R) 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:

$$Recall = \frac{TP}{TP \ + \ FN} = \frac{TP}{all\ ground\ truths}. \tag{3}$$

Average Precision (AP) 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:

$$AP = \sum_{n} (R_n - R_{n-1}) P_n, \tag{4}$$

where Pn and Rn are the precision and recall at the n-th threshold.

It is common to report two AP values: AP_{50} at a IoU threshold of 0.5 and $AP_{50:95}$ which averages AP across IoU thresholds from 0.5 to 0.95 in steps of 0.05. We will focus only on the second metric, since it offers a more comprehensive and stringent assessment.

Results

1. *Detection Performance*

All YOLO models trained for 150 epochs showed excellent detection results. All models show results on test data in the region of Precision ≈ 95%, Recall ≈ 95% and AP50:95 ≈ 84%, except for only YOLOv10-N. Detailed results of the quality testing are given in the table 3.

The YOLOv10 architecture, although having the relative low Recall, but high Precision on blue cell's detection. We can explain this by the fact that YOLOv10 is architecturally highly selective, which ensures high accuracy in measuring the counting sensitivity. Also, in our opinion, detecting blue cells was more difficult than red ones, due to the fact that there are significantly more of them and they can have different and more complex variations.

A high value for both Precision and Recall indicates that the model is not only good at identifying true positives (predictions of which classes are actually correct), but that it does so with a minimum number of errors, with virtually no positive cases missed. This means that the model is both accurate and sensitive, demonstrating balanced and high-quality behaviour in the detection task.

Table 3. **Performance evaluation on test data**

Model	Blue Cells			Red Cells		
	P (%)	R (%)	$AP_{50:95}$ (%)	P (%)	R (%)	$AP_{50:95}$ (%)
YOLOv8-N	95.2	96.2	85.2	96.2	95.1	83.8
YOLOv9-T	96.2	95.9	85	96.2	94.3	84.4
YOLOv10-N	97	91.8	84.4	94.3	94.4	83.6
YOLOv11-N	96.3	96.1	85.4	95.8	95	83.7
YOLOv12-N	96.4	95.5	85.5	96.5	95.5	84.2

It is worth noting that we used only the smallest versions of the models (nano or tiny). Considering that these networks are usually designed for devices with very limited computing power and usually do not imply the best quality, they showed excellent results. We also tried training larger models, but we could get either insignificant growth or even worse results. As a rule, larger models require more training data, which is not always possible. That is why we settled on the smaller versions of YOLO models.

2. *Dataset Scalability*

In the next experiment, we want to test the ability of YOLO models to learn on different amounts of training data. To do this, we vary the number of images in the training set and also fix the validation set (tab. 4). This experiment may be especially interesting because in biomedical applications, labelled datasets may be rare or labelling them may be difficult and time-consuming [21].

Table 4. **Comparative experiments on the scalability of a dataset for testing the learnability of models**

Experiment	Train (images)	Valid (images)
2%	5	72
5%	12	
10%	25	
25%	63	
50%	126	
100%	252	

We visualize the change in such quality metrics as Recall, Precision and AP50:95 over the epochs. An exponential moving average with alpha = 0.6 is pre-performed to smooth the curves (on fig. 3).

Of course, our experiment is not entirely fair, since in a real-world scenario a small amount of labelled data would be divided into a train-valid-test split (or train-valid). But still, we change the amount of available data in the training sample, and despite the extremely meagre sample, the neural network remains capable of learning and generalization, and can achieve relatively high quality results. Ultimately, only the training data has a direct impact on network learning and gradient propagation. The small size of the training sample means low diversity and variability of the data, which, however, does not interfere with the training and generalization of the network.

This is similar to the one-shot or few-shot learning technique, but this description is more applicable to some representative object of the class. In our case, several dozen or hundreds of class objects are located in one image. The smallest dataset size that we vary is 2% (or only 5 images out of 252), and in this sense it can be designated as few-shot image learning.

Conclusions

This study demonstrates the potential of YOLO-based object detection models for cell localization and classification in live-dead cell viability assays. We introduced a fully annotated, domainspecific dataset that enables researchers to reframe cell viability analysis as an object detection task. The results of our experiments demonstrate the capacity of YOLO models to learn meaningful representations and generalize effectively even with a limited amount of training data, utilizing as little as (2-10% of the total dataset. Notably, the latest

YOLO models (v8 to v12) consistently achieved high-quality performance metrics, with Precision ≈ 95%, Recall ≈ 95% and AP50:95 ≈ 84%, underscoring the robustness of these architectures for biomedical applications.

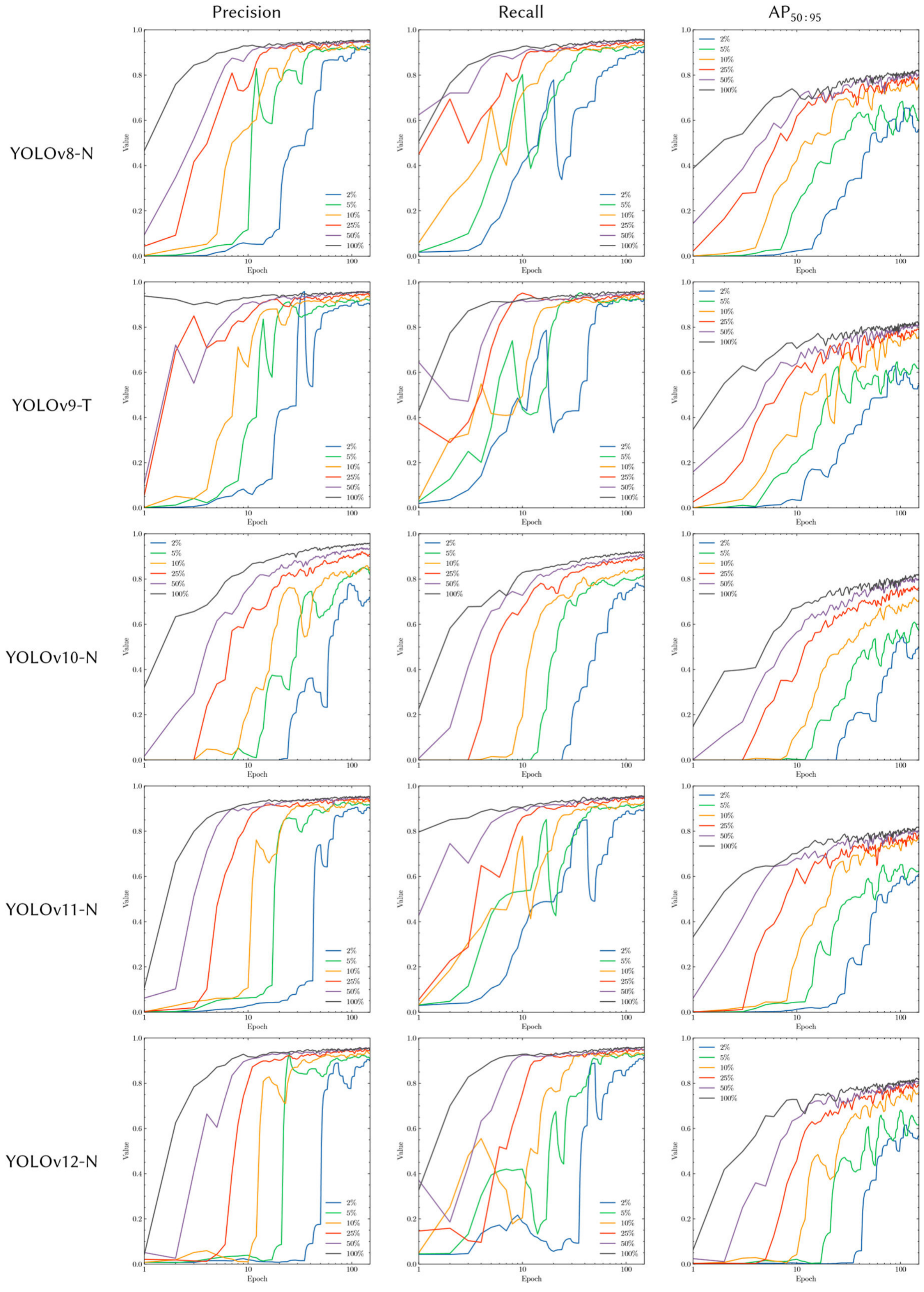


Figure 3. Validation metric curves. Each row presents the model validation curves. To smooth the curves, an exponential moving average with alpha = 0.6 is performed (logarithmic scale on x-axis)

Moreover, the models exhibit strong adaptability to small training samples, a critical feature for practical implementations where annotated data is often scarce. The introduction of this dataset, coupled with the demonstrated potential of modern YOLO architectures in this context, lays the foundation for further exploration of deep learning approaches in cell viability assays.

References

1. B. Aldughayfiq, F. Ashfaq, N. Z. Jhanjhi, M. Humayun, Yolov5-fpn: A robust framework for multi-sized cell counting in fluorescence images, Diagnostics 13 (2023) 2280. http://dx.doi.org/10.3390/diagnostics13132280.

2. C. Hu, S. He, Y. J. Lee, Y. He, E. M. Kong, H. Li, M. A. Anastasio, G. Popescu, Livedead assay on unlabeled cells using phase imaging with computational specificity, Nature Communications 13 (2022). http://dx.doi.org/10.1038/s41467-022-28214-x.

3. S. Lypez Flyrez, A. Gonzбlez-Briones, G. Hernбndez, C. Ramos, F. de la Prieta, Automatic cell counting with yolov5: A fluorescence microscopy approach, International Journal of Interactive Multimedia and Artificial Intelligence 8 (2023) 64. http://dx.doi.org/10.9781/ ijimai.2023.08.001.

4. M. Lu, W. Shi, Z. Jiang, B. Li, D. Ta, X. Liu, Deep learning method for cell count from transmitted-light microscope, Journal of Innovative Optical Health Sciences 16 (2023). http://dx.doi.org/10.1142/S1793545823500049.

5. Q. Lin, G. Ye, J. Wang, H. Liu, Roboflow: a data-centric workflow management system for developing ai-enhanced robots, in: A. Faust, D. Hsu, G. Neumann (Eds.), Proceedings of the 5th Conference on Robot Learning, volume 164 of Proceedings of Machine Learning Research, PMLR, 2022, pp. 1789–1794. https://proceedings.mlr.press/v164/lin22c.html.

6. 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, p. 779788. http://dx.doi.org/10.1109/cvpr.2016.91.

7. J. Terven, D.-M. Cyrdova-Esparza, J.-A. Romero-Gonzбlez, A comprehensive review of yolo architectures in computer vision: From yolov1 to yolov8 and yolo-nas, Machine Learning and Knowledge Extraction 5 (2023) 16801716. http://dx.doi.org/10.3390/make5040083.

8. M. L. Ali, Z. Zhang, The yolo framework: A comprehensive review of evolution, applications, and benchmarks in object detection, Computers 13 (2024) 336. http://dx.doi.org/10.3390/computers13120336.

9. M. Yaseen, What is yolov8: An in-depth exploration of the internal features of the nextgeneration object detector, 2024. https://arxiv.org/abs/2408.15857.

10. C.-Y. Wang, I.-H. Yeh, H.-Y. Mark Liao, YOLOv9: Learning What You Want to Learn Using Programmable Gradient Information, Springer Nature Switzerland, 2024, p. 121. http://dx.doi.org/10.1007/978-3-031-72751-1_1.

11. A. Wang, H. Chen, L. Liu, K. Chen, Z. Lin, J. Han, G. Ding, Yolov10: Real-time end-toend object detection, 2024. https://arxiv.org/abs/2405.14458.

12. R. Khanam, M. Hussain, Yolov11: An overview of the key architectural enhancements, 2024. https://arxiv.org/abs/2410.17725.

13. Y. Tian, Q. Ye, D. Doermann, Yolov12: Attention-centric real-time object detectors, 2025. https://arxiv.org/abs/2502.12524.

14. G. Jocher, J. Qiu, A. Chaurasia, Ultralytics yolo, 2025. https://doi.org/10.5281/zenodo.15514468.

15. T. Dao, D. Y. Fu, S. Ermon, A. Rudra, C. R´e, FlashAttention: Fast and memory-efficient exact attention with IO-awareness, in: Advances in Neural Information Processing Systems (NeurIPS), 2022.

16. T. Dao, FlashAttention-2: Faster attention with better parallelism and work partitioning, in: International Conference on Learning Representations (ICLR), 2024.

17. I. Loshchilov, F. Hutter, Decoupled weight decay regularization, 2017. https://arxiv. org/abs/1711.05101.

18. T. Kumar, R. Brennan, A. Mileo, M. Bendechache, Image data augmentation approaches: A comprehensive survey and future directions, IEEE Access 12 (2024) 187536187571. http://dx.doi.org/10.1109/ACCESS.2024.3470122.

19. A. Buslaev, V. I. Iglovikov, E. Khvedchenya, A. Parinov, M. Druzhinin, A. A. Kalinin, Albumentations: Fast and flexible image augmentations, Information 11 (2020). https://www.mdpi.com/2078-2489/11/2/125.

20. J. Terven, D.-M. Cordova-Esparza, J.-A. Romero-Gonzбlez, A. Ramнrez-Pedraza, E. A. Chбvez-Urbiola, A comprehensive survey of loss functions and metrics in deep learning, Artificial Intelligence Review 58 (2025). http://dx.doi.org/10.1007/s10462-025-11198-7.

21. J. Li, G. Zhu, C. Hua, M. Feng, B. Bennamoun, P. Li, X. Lu, J. Song, P. Shen, X. Xu, L. Mei, L. Zhang, S. A. A. Shah, M. Bennamoun, A systematic collection of medical image datasets for deep learning, ACM Computing Surveys 56 (2023) 151. http://dx.doi.org/10.1145/3615862.