Computer Methods and Programs in Biomedicine 286 (2026) 109602

Contents lists available at ScienceDirect

Computer Methods and Programs in Biomedicine

journal homepage: https://www.sciencedirect.com/journal/computer-methods-and-programs-in-biomedicine

SpheroSeg: Advancing tumor spheroid analysis through open-source deep learning☆

Michal Průšek [a,b], Adam Novozámský [a,*], Jan Škubník [c], Vladimíra Svobodová Pavlíčková [c], Silvie Rimpelová [c,*]

[a] The Czech Academy of Sciences, Institute of Information Theory and Automation, Pod Vodárenskou věží 4, Prague, 182 00, Czechia
[b] Czech Technical University in Prague, Faculty of Nuclear Sciences and Physical Engineering, Břehová 7, Prague, 115 19, Czechia
[c] University of Chemistry and Technology Prague, Department of Biochemistry and Microbiology, Technická 5, Prague, 166 28, Czechia

ARTICLE INFO

Keywords:
Tumor spheroids
3D cell culture
Deep learning segmentation
Quantitative image analysis
Bright-field microscopy

ABSTRACT

Background and Objective: Three-dimensional tumor spheroids are widely used *in vitro* models for studying tumor growth, invasion, and treatment response. Quantitative analysis remains challenging because imaging conditions vary across experiments, annotation policies differ between datasets, and large expert-corrected segmentation datasets are limited. This study introduces SpheroSeg, an open-source platform for bright-field spheroid segmentation that combines a large annotated dataset, standardized model benchmarking, and an accessible web-based analysis workflow.

Methods: We introduced SpheroHQ, a dataset comprising 22,683 bright-field images with expert-corrected annotations across seven cancer cell lines, and SpheroMix, a 32,367-image corpus integrating SpheroHQ with external spheroid datasets. Eight deep-learning segmentation architectures, including convolutional, attention-based, and state-space designs, were evaluated under a unified two-stage protocol and assessed on three stratified test sets: within-distribution SpheroHQ images, externally annotated DTS images, and the held-out out-of-distribution HTS-Seg dataset. Performance was assessed using standard segmentation metrics, with bootstrap 95% confidence intervals reported in the full benchmark to separate within-dataset performance, annotation-policy effects, and out-of-distribution generalization. A complementary SegFormer analysis was performed under the same stratified evaluation protocol to assess transformer-based segmentation models.

Results: Within the main eight-architecture benchmark, the best-performing models differed across test settings. CBAM-ResUNet achieved the highest within-distribution SpheroHQ IoU (0.9478), whereas MambaBot-UNet achieved the highest externally annotated DTS IoU (0.9351) and the strongest held-out HTS-Seg IoU (0.5867) within the main benchmark. The stratified analysis revealed a substantial gap between within-distribution and held-out out-of-distribution performance, indicating that the released models should be interpreted within the documented imaging and annotation conditions. In the complementary transformer analysis, SegFormer-B0 provided a fast transformer option with strong externally annotated DTS performance (IoU = 0.9449) and competitive HTS-Seg performance (IoU = 0.5139). The deployed web application provides three complementary model options: CBAM-ResUNet, SegFormer-B0, and MambaBot-UNet.

Conclusions: SpheroSeg contributes a large expert-corrected spheroid dataset, a protocol-aware benchmark, and a hosted and Dockerized web application for GPU-accelerated spheroid segmentation, polygon-based correction, morphometric measurement, and export for downstream analysis. Rather than serving as a universal spheroid detector, SpheroSeg provides a reproducible reference benchmark and deployable analysis platform for the documented bright-field imaging and annotation settings.

1. Introduction

Three-dimensional (3D) multicellular spheroids have become increasingly important *in vitro* models in cancer research and drug discovery. Unlike conventional two-dimensional (2D) monolayer cell cultures, which lack spatial complexity and have limited physiological relevance, 3D cell culture models more accurately replicate critical tumor microenvironment characteristics — including nutrient and oxygen gradients, cellular heterogeneity, and the formation of hypoxic cores. These characteristics are critical for accurately modeling tumor behavior, drug penetration, and therapeutic resistance in preclinical settings.

☆ This document is the result of the grant GA25-15933S funded by the Czech Science Foundation.
* Corresponding authors.
E-mail addresses: novozamsky@utia.cas.cz (A. Novozámský), silvie.rimpelova@vscht.cz (S. Rimpelová).

https://doi.org/10.1016/j.cmpb.2026.109602
Received 22 January 2026; Received in revised form 3 August 2026; Accepted 6 August 2026
Available online 14 August 2026
0169-2607/© 2026 The Authors. Published by Elsevier B.V. This is an open access article under the CC BY license (http://creativecommons.org/licenses/by/4.0/).

Spheroids are widely employed for simulating tumor growth and invasion dynamics, as well as in high-throughput drug screening assays. They are among the most effective models for toxicity assessment, provide a realistic representation of small avascular tumors *in vitro*, and serve as powerful tools for studying *in vivo* tissue behavior *in vitro*.

Quantitative analysis of spheroid properties, such as size, morphology, volume, diameter, surface area, perimeter, and invasive behavior, is crucial for assessing experimental outcomes. However, manual quantification of these parameters is time-consuming and labor-intensive, which presents a significant challenge, especially for large-scale or high-throughput screening experiments. This limitation has increased the demand for automated image analysis methods that can efficiently process large datasets.

Early approaches to spheroid image analysis often relied on classical image processing techniques implemented within general-purpose software platforms such as ImageJ/FIJI [1–7]. A variety of macros and plugins were developed for these platforms to automate basic measurements, including spheroid size, shape, and diameter [5–7]. Over time, several specialized software tools were developed, some built upon or designed to interact with ImageJ/FIJI. Examples include PCaAnalyser, a Java-based tool developed to analyze spheroids of prostate cancer cells [1]; AnaSP, a MATLAB-based suite for morphological analysis [8]; and INSIDIA, a FIJI macro focused on invasion analysis, which was later updated in INSIDIA 2.0 [9,10]. Other tools, such as SpheroidSizer, offer user-friendly interfaces to estimate the spheroid size [3], and TASI, which enables spatiotemporal quantification of spheroid behavior [5]. Standalone applications such as AMIDA have also been introduced to support large-scale 3D culture screening experiments [2]. These tools typically provide quantitative metrics such as area, diameter, perimeter, and volume, and some extend their functionality to include the evaluation of invasive behavior. In parallel, commercial systems such as Opera [1] and specialized imaging cytometers such as Celigo [11] have incorporated automated analysis capabilities, often as components of integrated high-throughput screening platforms.

However, methods relying solely on classical image processing or fixed parameter settings often encounter substantial limitations. These approaches are typically sensitive to variations in spheroid morphology, size, texture, and image quality and may struggle with common issues such as uneven illumination or the presence of debris [2]. Quantifying complex cellular behaviors, especially invasion, presents additional challenges, especially when cell types do not form a well-defined invasive front, necessitating more advanced metrics than simple area-based measurements [12]. Moreover, achieving consistent performance across different cell lines and experimental setups often requires extensive manual parameter tuning, limiting both the scalability and robustness of such techniques [2].

More recently, deep learning techniques have gained significant traction in biomedical image analysis due to their ability to automatically learn complex patterns and features, offering notable improvements in segmentation accuracy and robustness [13–16]. An early pioneering work in this direction was by Sadanandan et al. [17], who demonstrated the potential of deep adversarial networks for spheroid segmentation, establishing a foundation for subsequent developments. Architectures such as U-Net [18] and Fully Convolutional Networks (FCNs) [19] have been widely adopted for semantic segmentation tasks, while models like Mask R-CNN [20] are increasingly used for instance segmentation [6,15,16]. Several deep learning-based tools have been developed specifically for spheroid analysis, including SpheroidJ [15] and SpheroScan [16], which use neural networks for accurate segmentation of 3D spheroids. The SMART system [21] further advances this trend by incorporating a refined convolutional neural network (CNN) for automated spheroid detection and analysis while also introducing novel quantitative metrics such as the Excess Perimeter Index (EPI) and Multiscale Entropy Index (MSEI) to characterize invasive behavior better. Beyond analytical applications, artificial intelligence has also been employed for physical manipulation tasks. For instance, SpheroidPicker utilizes deep learning to facilitate morphology-based selection and automated transfer of spheroids, streamlining experimental workflows [6]. A recently published study by [22] introduced the HTS-Seg dataset, a collection of drug-treated heterotypic spheroids imaged in a microfluidic chip system. The authors provide 95 images and 2980 tiled patches with semi-automated annotations and benchmark several deep-learning models, with DeepLab achieving the highest IoU (91.17%). While valuable as a dataset resource for a specific experimental setup, HTS-Seg does not aim to provide a ready-to-use segmentation tool or a deployment-oriented benchmark.

Despite the advances brought about by deep learning, several challenges remain. One of the most prominent issues is the domain shift problem [23,24], where models trained on data from a specific source, such as variations in microscopy setup, staining, or cell type, often perform poorly when applied to images from different experimental settings. To address this, recent studies have explored the use of style transfer techniques, which aim to transform images from a new domain to match the visual characteristics of the training data [24]. A related challenge is the dependence of supervised segmentation methods on large expert-annotated datasets. Self-supervised contrastive learning offers a strategy for reducing this dependence by learning transferable visual representations from unlabeled images. A recent study on senescent-cell bright-field microscopy demonstrated the utility of combining contrastive pretraining with an instance-segmentation framework under limited labeled-data conditions [25]. Although that study addresses a biologically distinct application rather than tumor spheroids, it provides relevant methodological evidence for annotation-efficient microscopy segmentation. Furthermore, while deep learning models can achieve highly accurate segmentation, extracting meaningful biological insights still requires robust post-processing, including comprehensive parameter extraction and interpretation [10]. In summary, automated analysis of spheroid images has progressed significantly, evolving from classical image processing methods to sophisticated deep learning-based approaches.

Objective of This Study. The objective of this study is to develop, evaluate, and deploy an open-source segmentation pipeline for bright-field tumor-spheroid microscopy that addresses three specific gaps in the current literature: the limited availability of large expert-corrected public datasets, the lack of protocol-aware evaluation across annotation policies and external test sets, and the limited accessibility of trained models through deployable end-user software. Rather than claiming universal generalization, we explicitly quantify performance under within-distribution, externally annotated, and out-of-distribution test conditions.

Contributions. Within this scope, the main contributions of this work are:

(a) SpheroHQ, an expert-corrected bright-field tumor-spheroid dataset containing 22,683 images from seven cancer cell lines, and SpheroMix, a 32,367-image corpus integrating SpheroHQ with SLiMIA, the DTS training split, and SpheroidJ;
(b) a strict cell-line-disjoint split of SpheroHQ and a stratified evaluation design that reports performance separately on within-distribution SpheroHQ, externally annotated DTS, and held-out HTS-Seg data;
(c) a standardized benchmark of eight deep-learning segmentation architectures, corresponding to 16 pretrained and finetuned configurations, with bootstrap 95% confidence intervals reported for the principal segmentation metrics;
(d) a completed four-annotator re-annotation pilot on a 150-image subset of the DTS test set, used to quantify the annotation-policy gap between SpheroHQ-style inclusive labels and the original DTS core-only ground truth;

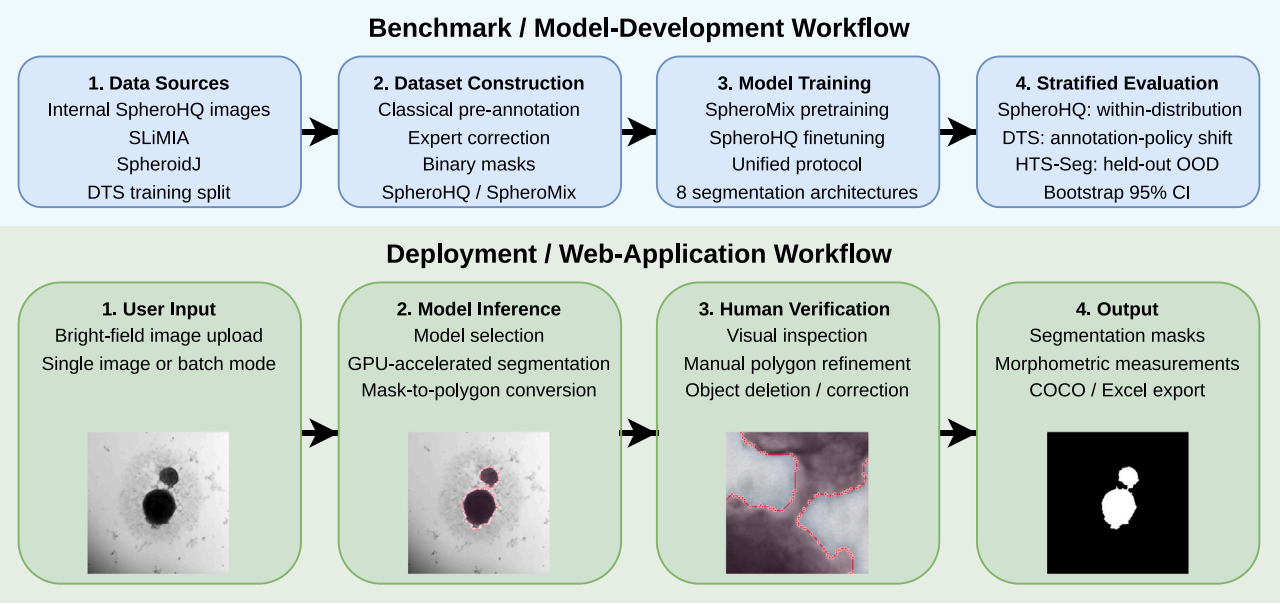

Fig. 1. Overview of the SpheroSeg workflow. The **upper row** summarizes the benchmark and model-development pipeline. Internal bright-field spheroid images and external training sources are assembled into SpheroHQ and SpheroMix, used for standardized two-stage deep-learning training, and evaluated separately on SpheroHQ within-distribution data, externally annotated DTS data, and held-out HTS-Seg data. The **lower row** summarizes the deployment pipeline in the SpheroSeg web application: users upload single images or batches, select a deployed model for GPU-accelerated segmentation, optionally verify and refine the resulting polygons, and export segmentation masks, morphometric measurements, and annotation files for downstream analysis.

(e) a complementary SegFormer transformer analysis [26], including SpheroMix-pretrained and SpheroHQ-finetuned B0, B2, and B4 variants, used to assess whether cross-dataset behavior depends on the architecture family and training protocol;
(f) a hosted and Dockerized open-source web application that deploys three complementary models, CBAM-ResUNet, SegFormer-B0, and MambaBot-UNet, and provides automated segmentation, manual polygon refinement, morphometric measurements, and export for downstream analysis.

Fig. 1 summarizes the full SpheroSeg workflow, from dataset construction and standardized model training to stratified evaluation and web-based deployment.

2. Methods

2.1. Biological data

Cell Maintenance and Passaging. In this study, the following human cell lines were used: cells from non-small cell lung adenocarcinoma (A549; CLS, Germany), triple-negative breast carcinoma (UFH-001, Merck, USA), pancreatic adenocarcinoma (BxPC-3 and MIA PaCa-2; ATCC, USA), cervical carcinoma (HeLa, ATCC, USA), prostatic and breast carcinoma (PC-3 and MCF-7; ATCC and Merck, respectively, USA). Cell lines were kept in the following cultivation media: A549, UFH-001, PC-3, MCF-7, MIA PaCa-2, and HeLa in high-glucose Dulbecco's modified Eagle medium (DMEM) supplemented with 10% (*v/v*) fetal bovine serum (FBS; Merck, USA) and 2 mM L-glutamine (Merck, USA), and BxPC-3 in RPMI-1640 medium + 10% (*v/v*) FBS + 2 mM L-glutamine. Cells were kept in standard cultivation conditions at 37 °C, 5% CO_2 in the atmosphere and 95% humidity. The cells were regularly passaged (2–3 times per week) and kept in the exponential growth phase. Cell passaging was carried out as follows: cells that grow in a monolayer (two-dimensional culture) in a 10 cm diameter cell culture plate (Avantor, Czech Republic) were rinsed with 4 ml of sterile phosphate-buffered saline (PBS, pH 7.4), detached with 1 ml of 0.25% (*v/v*) trypsin with 0.02% (*v/v*) ethylenediaminetetraacetic acid in Hank's balanced salt solution for 2–10 min. (cell line-dependent) and seeded in new cell culture plates containing fresh cell culture media.

Spheroid Preparation and Imaging. Cancer cell spheroids, three-dimensional cell culture models, were prepared from a cell suspension obtained during the passaging of cells previously grown in a monolayer culture. Cell spheroids were formed using Nunclon™ Sphera™ 96-well clear, round U-bottom plates with an ultra-low-attachment surface (Thermo Fisher Scientific, USA), similarly as reported in Ref. [27]. Briefly, specific numbers of cells were seeded into each well in 100 µL of appropriate cell culture media: 2000 A549, 3000 UFH-001, 300 MIA PaCa-2, 4000 BxPC-3, 1500 PC-3, 3000 MCF-7, and 500 HeLa cells. The numbers of seeded cells were optimized prior to the production of cancer cell spheroids to achieve a diameter of approximately 400 µm after 96 h of incubation — considered optimal for potential subsequent drug testing assays [28]. Plates were incubated under standard conditions to enable spheroid formation, with one spheroid forming per well that would be suitable for individual image analysis or treatment. Although typically a single spheroid forms per well, drug treatment may lead to partial disintegration into multiple fragments.

Cancer cell spheroids were initially imaged after 4 days of incubation and subsequently every 2–3 days for up to 17 days using live-cell bright-field microscopy. Imaging was performed using inverted Olympus IX-81 and Olympus IX-83 microscopes (Olympus, Japan) operated with xCellence and cellSens software, respectively. The systems were equipped with an EM-CCD camera C9100-02 (Hamamatsu, Germany) or Orca Flash 4.0 (Hamamatsu, Germany). Two dry objectives were used for the imaging: a 4× objective with a numerical aperture of 0.1 and a 10× objective with a numerical aperture of 0.3 (both Olympus, Japan), providing total magnifications of 40 and 100×, respectively. Images of individual spheroids (one per well) were saved in TIFF format for further analysis.

2.2. Image segmentation pipeline

The initial phase of this research focused on developing robust spheroid segmentation using classical image processing techniques. These methods were used to generate a preliminary segmentation

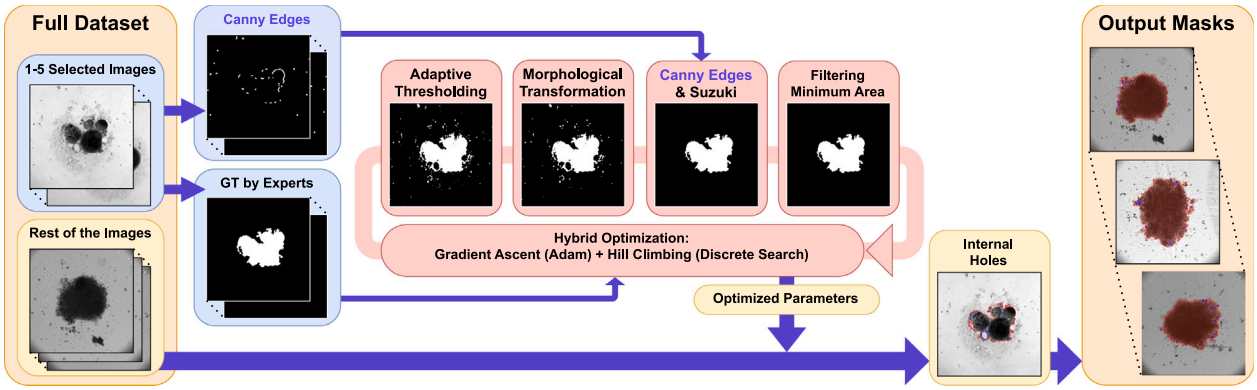

Fig. 2. The classical segmentation pipeline combines adaptive thresholding, Canny edge detection with Suzuki contour extraction, and morphological transformations. A hybrid optimization scheme — combining gradient-based search (Adam optimizer on a numerically estimated IoU objective) and discrete hill climbing — was applied to a small set of manually reviewed samples (1–5 per case) to refine parameters such as thresholds and filter criteria (e.g., minimum area, internal hole removal).

of spheroids, which served as a basis for expert refinement. Using the characteristic intensity differences between the spheroids and the typically non-uniform background, the pipeline was configured to produce sufficiently accurate contours. This approach enabled experts to focus on refining boundary inaccuracies instead of performing manual annotations from scratch, substantially accelerating the annotation workflow.

After evaluating various regional segmentation methods on grayscale images, three adaptive thresholding techniques were selected for further development: Gaussian adaptive thresholding, Sauvola method [29], and Niblack method [30]. Global thresholding methods such as Otsu's method [31] proved inadequate due to uneven illumination across the field of view. These adaptive thresholding methods compute a local threshold for each pixel based on the statistical properties of its neighborhood. Gaussian adaptive thresholding uses a Gaussian-weighted mean, while Sauvola's and Niblack's methods rely on the local mean and standard deviation.

The binary masks generated by the adaptive thresholding methods were further refined to enhance segmentation accuracy. First, a morphological opening was applied to remove noise and smooth boundary irregularities, and then the Suzuki algorithm [32] was used to extract the contours. A crucial step in the refinement process involved applying the Canny edge detector [33] on the original grayscale image. Only the contours that intersected with the Canny edges were retained. The binary masks obtained from the thresholding stage were then refined. To better capture the often ambiguous edges of the spheroids, a morphological dilation was applied. Only objects (contours) whose area exceeds a certain minimum threshold were retained. The tool also identifies and removes images that are damaged or irrelevant (e.g., scratches, fibers, or partial capture). The segmentation of holes inside the spheroids proceeds as follows: the binary spheroid mask is dilated, inner contours are extracted, and an eroded inner contour mask is subtracted from the main spheroid mask.

Parameter Optimization. Crucially, the parameters for these algorithms (such as thresholding windows, sensitivity constants k, and kernel sizes) were not chosen arbitrarily. We employed a hybrid optimization scheme to maximize the Intersection over Union (IoU) on a small set of representative reference images (typically 1–5 per dataset). This scheme combined a gradient-based search (using the Adam optimizer on a numerically estimated IoU objective) for continuous variables and discrete hill climbing for integer-based parameters. This automated tuning ensured that the classical pipeline adapted robustly to the specific visual characteristics of each cell line before generating the large-scale pre-annotations. Fig. 2 illustrates this pipeline.

2.3. Deep learning architectures

We evaluated eight deep-learning segmentation architectures selected for complementary architectural properties and practical deployment relevance. The benchmark includes established segmentation baselines and three architecture variants implemented in this work by integrating established components: LC-ResUNet, MA-ResUNet, and MambaBot-UNet. MambaBot-UNet combines a standard U-Net backbone with a Mamba-based bottleneck, whereas LC-ResUNet and MA-ResUNet integrate lightweight and multi-attention mechanisms into ResUNet-style encoder–decoder architectures. All architectures process 1024 × 1024 pixel images and produce binary segmentation masks at the same resolution. The three architecture variants implemented in this work are visualized in Fig. 3.

U-Net: Standard Implementation

We implemented the classical U-Net architecture [18] with five levels using channels [64, 128, 256, 512, 1024]. Each level consists of double convolution blocks with instance normalization and ReLU activation. The architecture includes optional deep supervision for improved gradient flow.

HRNet: High-Resolution Network

HRNet [34] maintains high-resolution representations throughout processing via parallel multi-resolution streams. Starting with a single stream, it progressively adds parallel branches at 1/2×, 1/4×, and 1/8×resolutions with channels [48, 96, 192, 384]. Cross-resolution exchange units enable information fusion between streams.

PSPNet: Pyramid Scene Parsing Network

PSPNet [35] employs a ResNet-101 backbone with dilated convolutions (output stride 8), followed by pyramid pooling at scales [1 × 1, 2 × 2, 3 × 3, 6 × 6]. We replaced batch normalization with group normalization for stability.

CBAM-ResUNet: ResUNet with CBAM Attention

CBAM-ResUNet integrates Convolutional Block Attention Module [36] into a ResUNet backbone. The network uses feature channels of [64, 128, 256, 512] with a 1024-channel bottleneck. CBAM applies sequential channel and spatial attention.

MA-ResUNet: Multi-Attention ResUNet

MA-ResUNet extends the ResUNet framework [37] with three complementary attention mechanisms. Using feature channels [32, 64, 128, 256] and a 512-channel bottleneck, it integrates: (1) SimAM [38]; (2) TripletAttention [39]; and (3) LightweightSelfAttention with 8 heads in the bottleneck. We also evaluated a memory-efficient variant, MA-ResUNet-Mini.

LC-ResUNet: Lightweight CBAM-ResUNet

LC-ResUNet implements a lightweight variant of CBAM attention optimized for robust generalization. With channels [48, 96, 192, 384,

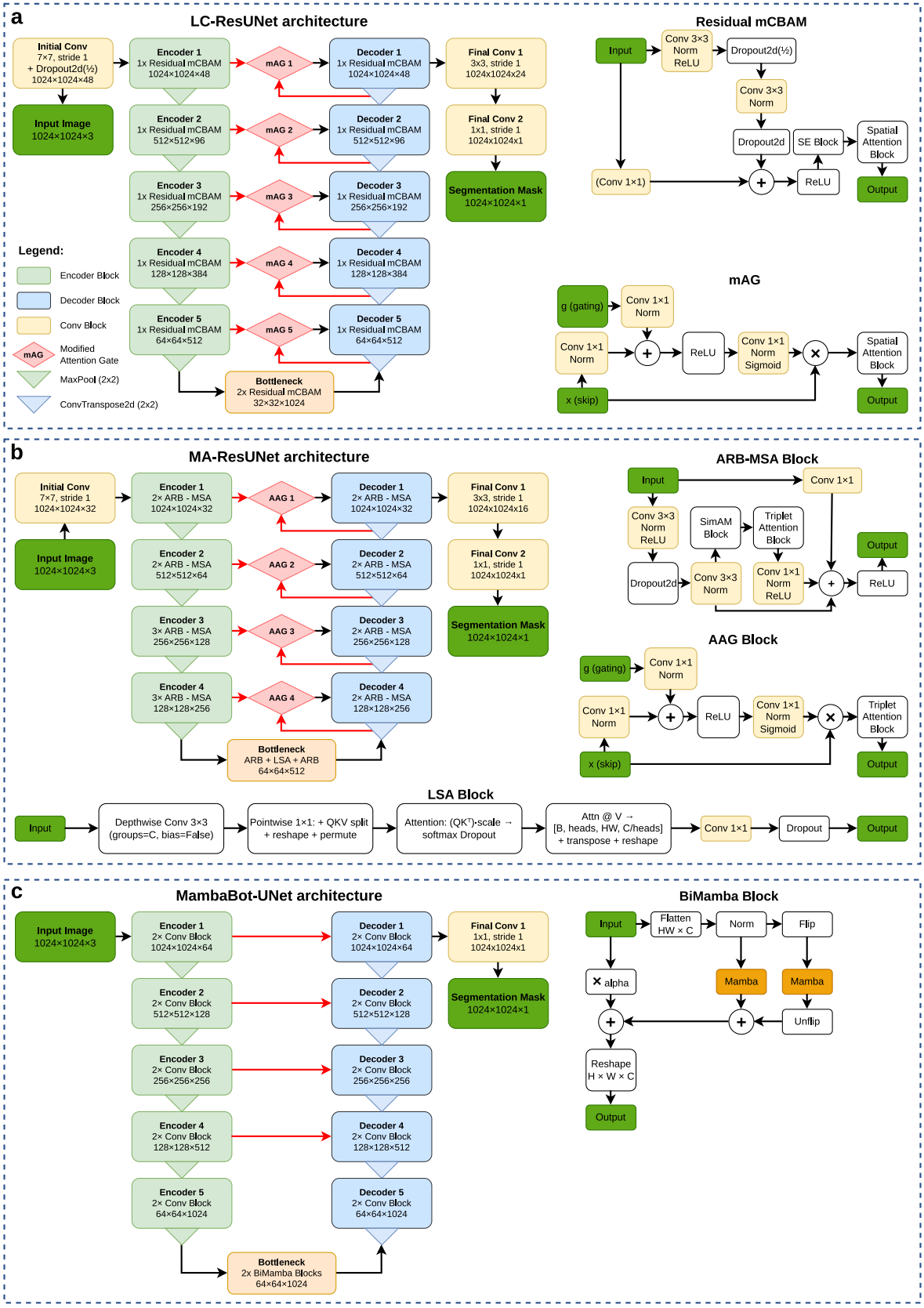

Fig. 3. Architecture variants implemented in this work. **a**, encoder–decoder *LC-ResUNet* with residual modified CBAM blocks and modified attention gates. **b**, encoder–decoder *MA-ResUNet* with multi-attention residual blocks and attention gates. **c**, encoder–decoder *MambaBot-UNet* with two BiMamba blocks in the bottleneck.

Table 1
Comparison of classical segmentation accuracy with existing tools.

Method	Median IoU (%)
Developed classical methods (prusek-spheroid)	
Sauvola thresholding	**96.56**
Niblack thresholding	96.43
Gaussian adaptive thresholding	96.40
Other benchmarked tools	
SpheroidJ [15]	95.92
μSAM [44]	93.80
SpheroScan [16]	92.66
AnaSP – Automatic [8]	83.44
AnaSP – Multiple objects [8]	83.44
AnaSP – VGG-16 [8]	39.66

Median IoU scores from 5-fold cross-validation on 72 bright-field images.

512], it employs simplified channel attention (SEBlock) using only global average pooling rather than both average and max pooling.

MambaBot-UNet: U-Net with a Mamba bottleneck

MambaBot-UNet is a U-Net variant that augments the bottleneck stage with state-space modeling. The encoder and decoder follow our standard five-level U-Net configuration with channels [64, 128, 256, 512, 1024], while the 64×64 bottleneck representation is processed by two stacked bidirectional Mamba blocks. Each block combines forward and flipped-sequence Mamba scans [40,41] with a learnable residual scaling parameter inspired by LightM-UNet [42]. The bottleneck-only placement follows UMambaBot [43].

3. Results

3.1. Performance of classical segmentation

In order to evaluate the performance of the developed classical segmentation methods, 5-fold cross-validation was performed on a manually annotated dataset of 72 bright-field images covering four spheroid models (PC-3, BxPC-3, A549, and MIA PaCa-2), as shown in Table 1. The three implemented methods consistently achieved high segmentation accuracy, with median IoU scores of 0.9656, 0.9643, and 0.9640, respectively. The effectiveness of these methods was further assessed through comparative evaluation against publicly available tools, including spheroid-specific software and μSAM (Segment Anything for Microscopy) [44], a microscopy-oriented implementation of the Segment Anything framework using a ViT-B backbone. The developed methods showed competitive median IoU on the evaluated 72-image benchmark.

3.2. Deep learning model performance

In the main benchmark, we evaluated 16 model configurations, corresponding to eight architectures and two training strategies. The evaluation was stratified across three test settings totaling 1115 images: the within-distribution SpheroHQ BxPC-3 test split ($n = 653$), the externally annotated DTS test split with a different core-only annotation policy ($n = 366$), and the held-out HTS-Seg dataset representing out-of-distribution acquisition conditions and cell lines ($n = 96$). This design separates within-distribution performance, annotation-policy effects, and out-of-distribution behavior rather than collapsing them into a single aggregate score.

Table 2 summarizes all configurations in the main eight-architecture benchmark under the standardized protocol. The best-performing model differed across test settings. On the SpheroHQ test split, CBAM-ResUNet (finetuned) achieved the highest IoU, reaching 0.9478 [0.944–0.952]. On the externally annotated DTS test split, MambaBot-UNet (finetuned) achieved the highest IoU, reaching 0.9351 [0.926–0.943]. On the held-out HTS-Seg dataset, MambaBot-UNet (finetuned) also achieved the strongest performance, with an IoU of 0.5867 [0.563–0.607]. MambaBot-UNet and HRNet were the fastest configurations, with similar offline inference times of 26 ms and 27 ms per 1024×1024 image, respectively.

3.3. SpheroSeg: Web-based analysis platform

As a practical outcome of this study, we developed SpheroSeg, an automated analysis platform designed to make the evaluated segmentation models accessible for routine laboratory use without requiring programming expertise. An example screenshot of the web interface is shown in the middle panel of Fig. 4. Representative segmentation outputs from the three deployed models are shown in Fig. 8. The examples compare CBAM-ResUNet, MambaBot-UNet, and SegFormer-B0 on the same DTS and SpheroHQ images, illustrating how the deployed options differ in practically relevant cases, including external annotation-policy mismatch and ambiguous spheroid boundaries.

Workflow and Features. The platform integrates three complementary deployed models into a user-friendly graphical interface, with their selection rationale, deployment details, and production performance measurements reported in Appendix A.10. The lower row of Fig. 1 summarizes the user-facing workflow: users upload bright-field images, select the desired model, and initiate automatic segmentation. Following the automated process, an integrated polygon editor allows researchers to verify and refine the results. This editor supports vertex manipulation, boundary refinement, and object splitting, ensuring that ambiguous cases can be corrected by a human expert. The application supports export to standard formats, including COCO JSON for annotations and spreadsheet-compatible tabular data for statistical analysis.

Morphometric Parameter Definitions. Beyond segmentation, the platform automatically computes a set of morphometric parameters for each identified spheroid. These include area (px^2), perimeter (px), equivalent diameter, circularity, solidity, and convexity; detailed definitions are provided in Appendix A.10. These metrics support quantitative characterization of spheroid morphology over time and across experimental conditions, while downstream biological interpretation remains application-dependent.

4. Discussion

Dataset, Benchmark, and Deployment Context. This work contributes three connected resources for bright-field spheroid analysis: the SpheroHQ expert-corrected dataset, the larger SpheroMix training corpus integrating external datasets, and the SpheroSeg web application for model deployment and morphometric analysis. Compared with classical tools such as AnaSP [8] and SpheroidJ [15], the proposed workflow shifts the main segmentation step from fixed-parameter image processing to trained deep-learning models. However, the role of classical image processing remains important: in this study, adaptive thresholding and contour-based methods were essential for generating pre-annotations that could then be refined by experts at scale.

The revised benchmark also shows that model ranking depends on the evaluation setting. CBAM-ResUNet achieved the highest within-distribution SpheroHQ IoU, whereas MambaBot-UNet achieved the strongest DTS and HTS-Seg results within the main eight-architecture benchmark. This supports reporting stratified results rather than a single aggregate score, especially when annotation policies and acquisition conditions differ across datasets. The complementary SegFormer analysis is interpreted separately from the main benchmark. It further illustrates that transformer-based models can show different cross-dataset behavior, particularly on the externally annotated DTS test set.

Annotation Policies and Interpretation of Cross-Dataset Performance. The main methodological lesson from the cross-dataset evaluation is that performance differences cannot be interpreted solely as segmentation errors. The SpheroHQ annotations follow an inclusive policy that captures the visible spheroid structure, including diffuse or degrading peripheral regions when they are judged to belong to the spheroid. In contrast, the original DTS labels follow a more conservative core-only policy. A model trained under the inclusive policy can

Table 2
Summary of model performance under the standardized evaluation protocol. Performance is reported separately for three test settings: the within-distribution SpheroHQ BxPC-3 test split ($n = 653$), the externally annotated DTS test split with a different core-only annotation policy ($n = 366$), and the held-out HTS-Seg dataset representing out-of-distribution acquisition conditions and cell lines ($n = 96$). Each IoU value is reported as the mean with a bootstrap 95% confidence interval computed from 10,000 resamples of per-image IoU scores.

Model	Train	SpheroHQ IoU [95% CI] ($n = 653$)	DTS IoU [95% CI] ($n = 366$)	HTS-Seg IoU [95% CI] ($n = 96$)	Time[a] (ms)
U-Net	Pretr.	0.9418 [.937,.946]	0.8820 [.867,.896]	0.4032 [.382,.424]	44
U-Net	Finet.	0.9393 [.933,.945]	0.9096 [.897,.921]	0.4205 [.397,.443]	44
HRNet	Pretr.	0.9250 [.919,.931]	0.9011 [.888,.913]	0.2600 [.222,.298]	27
HRNet	Finet.	0.9361 [.931,.941]	0.8691 [.856,.880]	0.1095 [.080,.140]	27
PSPNet	Pretr.	0.9287 [.922,.935]	0.9072 [.894,.919]	0.4343 [.411,.458]	47
PSPNet	Finet.	0.9395 [.935,.944]	0.8746 [.859,.889]	0.2475 [.212,.283]	47
CBAM-ResUNet	Pretr.	0.9475 [.944,.951]	0.8781 [.865,.890]	0.2239 [.201,.248]	200
CBAM-ResUNet	Finet.	**0.9478** [.944,.952]	0.8614 [.847,.874]	0.1744 [.146,.203]	200
Our architectures:					
MA-ResUNet	Pretr.	0.9404 [.936,.945]	0.9068 [.895,.917]	0.5442 [.521,.566]	93
MA-ResUNet	Finet.	0.9353 [.931,.940]	0.8962 [.884,.907]	0.4931 [.463,.523]	93
MA-ResUNet-Mini	Pretr.	0.9328 [.928,.937]	0.8917 [.879,.903]	0.2579 [.232,.284]	55
MA-ResUNet-Mini	Finet.	0.9385 [.934,.943]	0.8632 [.849,.876]	0.2075 [.186,.229]	55
LC-ResUNet	Pretr.	0.9312 [.925,.937]	0.9067 [.895,.917]	0.5507 [.536,.565]	58
LC-ResUNet	Finet.	0.9324 [.926,.938]	0.8877 [.876,.897]	0.4416 [.416,.467]	58
MambaBot-UNet	Pretr.	0.9138 [.906,.922]	0.9304 [.920,.939]	0.5364 [.518,.555]	**26**
MambaBot-UNet	Finet.	0.9238 [.916,.931]	**0.9351** [.926,.943]	**0.5867** [.563,.607]	**26**

[a] Per-image inference time measured on an NVIDIA L40S GPU with 48 GB VRAM using 1024 × 1024 input images, batch size 1, mixed-precision bfloat16 inference, and a 30-iteration warm-up to prime CUDA caches and kernel autotuning. These offline benchmark timings are not directly comparable to the production deployment latencies reported in Table 9, which were measured on a different GPU and serving stack.

therefore be penalized when evaluated against DTS even if it delineates biologically plausible spheroid material.

The completed DTS re-annotation pilot confirms this interpretation quantitatively. Four annotators working under the SpheroHQ inclusive policy showed high within-policy agreement. A simultaneous truth and performance level estimation (STAPLE) [45] consensus mask derived from their annotations achieved an IoU of approximately 0.92 against the original DTS ground truth. This value serves as an empirical cross-policy agreement reference for interpreting inclusive-policy annotations evaluated against the original DTS core-only labels. It is a descriptive agreement statistic rather than a strict upper bound on model performance, and model IoU values may therefore exceed it. The DTS results should be read as an annotation-policy-shift evaluation rather than as a conventional out-of-distribution benchmark.

Training Protocol and Specialization–Generalization Trade-Off. The main evaluated architectures were trained under a standardized protocol to improve comparability across models. This reduces the confounding effects of model-specific learning rates, batch sizes, epoch budgets, and encoder-freezing schedules. Under this unified protocol, the two-stage strategy has a clear interpretation: pretraining on SpheroMix exposes the models to broader imaging and annotation variability, whereas finetuning on SpheroHQ specializes them toward the expert-corrected inclusive SpheroHQ policy.

The complementary SegFormer transformer analysis further supports this interpretation. It includes SpheroMix-pretrained and SpheroHQ-finetuned B0, B2, and B4 variants evaluated under the same stratified test protocol. The results show that the effect of second-stage finetuning is not uniform across transformer scales: SegFormer-B4 benefited from SpheroHQ finetuning on both SpheroHQ and HTS-Seg, whereas SegFormer-B0 changed only marginally and SegFormer-B2 decreased on the external test sets. This indicates that cross-dataset behavior depends on both architecture family and training protocol, rather than on annotation-policy exposure alone. For use cases aligned with SpheroHQ-style annotations, the strongest models from the main benchmark remain appropriate; for settings with different annotation conventions or acquisition conditions, transformer-based alternatives, policy-aware training, or additional local finetuning may be beneficial.

Deployment Considerations. The SpheroSeg web application is intended to make the trained models usable in routine laboratory workflows without requiring programming expertise. The platform supports image upload, automated segmentation, manual polygon refinement, morphometric measurement, and export to standard formats. The final deployment set was selected to cover three complementary operating points: CBAM-ResUNet for highest SpheroHQ performance, SegFormer-B0 for fast transformer inference and efficient batch segmentation, and MambaBot-UNet for strongest external-test performance within the main benchmark.

These deployment results should be interpreted as technical workflow measurements rather than biological validation of a complete high-throughput screening pipeline. The application can support batch processing and downstream morphometric analysis, but the biological impact of segmentation differences on endpoints such as drug-response curves, IC50 estimates, or invasion phenotypes remains a separate validation question.

4.1. Limitations

Several limitations should be considered when interpreting and applying the results of this study.

Annotation-Policy Variability. The SpheroHQ, DTS, and HTS-Seg datasets do not use identical annotation policies. In particular, SpheroHQ follows an inclusive annotation policy, whereas DTS uses a more conservative core-only convention. We address this by reporting stratified results and by adding the DTS re-annotation pilot, but annotation-policy variability remains a central limitation for cross-dataset comparison.

Out-of-Distribution Performance. The held-out HTS-Seg benchmark shows a substantial drop relative to within-distribution SpheroHQ performance. This indicates that the released models should not be interpreted as universal spheroid detectors. They are best viewed as strong baselines for imaging and annotation conditions similar to those represented in SpheroHQ and SpheroMix.

Single-Center Bias of SpheroHQ. Although SpheroMix incorporates external datasets, the core SpheroHQ dataset was acquired in

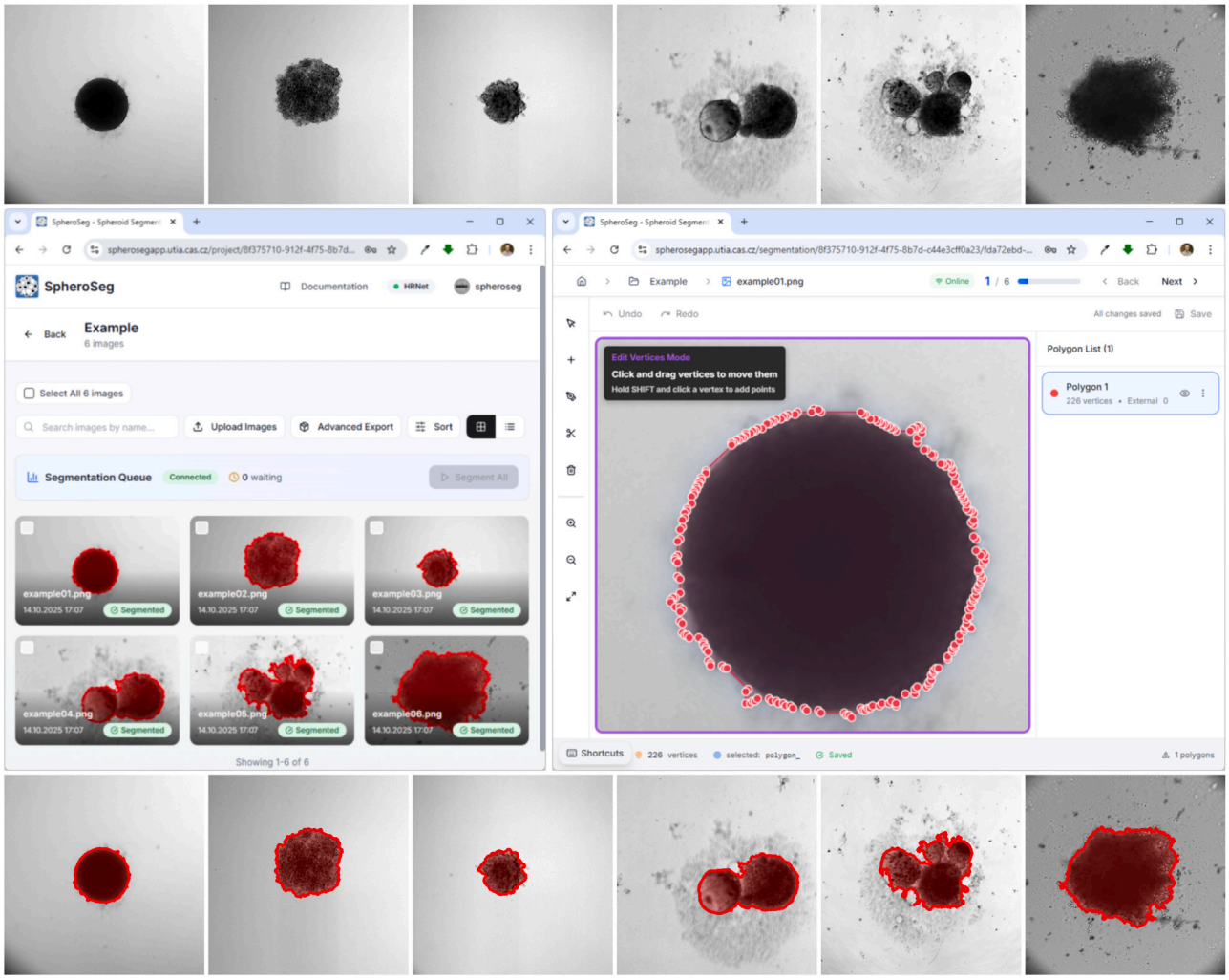

Fig. 4. Overview of the SpheroSeg web application. The central panel shows the browser-based graphical interface for spheroid segmentation, visualization, and analysis. The upper row provides representative bright-field spheroid images processed with the deployed CBAM-ResUNet model under different conditions (untreated – first three, compound-treated – last three). The bottom row shows the corresponding segmentation mask overlays in red.

a single laboratory and annotated under one expert-supervised policy. Cross-laboratory replication with independent acquisition systems and annotation teams is needed before claiming protocol-independent performance.

Modality and Dimensional Scope. All training and evaluation data in this study are two-dimensional bright-field images. The reported results do not establish performance for fluorescence microscopy, phase-contrast imaging, multimodal bright-field/fluorescence overlays, or 3D z-stack acquisitions. Extending the method to these settings would require additional annotated data or domain-adaptation strategies.

Bounded DTS Re-Annotation Pilot. The re-annotation pilot covers 150 of the 366 images in the DTS test set and should therefore be interpreted as a bounded estimate of the annotation-policy gap. The resulting IoU of 0.9201 summarizes the observed cross-policy agreement on this subset and should not be interpreted as a strict upper bound on model performance. A full re-annotation of the DTS test partition, together with models that explicitly account for both annotation policies, remains a valuable direction for future work.

Computational Requirements. The reported interactive latency assumes GPU-equipped infrastructure. CPU-only inference is possible but is not expected to match the reported production throughput. Users without suitable GPU resources may need to rely on hosted deployment or batch-mode processing.

5. Conclusions

We present SpheroSeg, an open-source platform for bright-field tumor-spheroid segmentation that combines expert-corrected data, standardized model evaluation, and deployable software. The study releases SpheroHQ, a dataset of 22,683 expert-corrected images across seven cancer cell lines, and SpheroMix, a 32,367-image corpus integrating SpheroHQ with external spheroid datasets. Classical image-processing methods were used to support scalable pre-annotation, while deep-learning models were trained and evaluated for automated segmentation.

Our evaluation emphasizes stratified reporting. Within-distribution performance was assessed on the SpheroHQ BxPC-3 test split, annotation-policy effects were examined on the externally annotated DTS test split, and broader out-of-distribution behavior was tested on the held-out HTS-Seg dataset. The completed DTS re-annotation

pilot shows that part of the measured SpheroHQ–DTS gap reflects annotation-policy mismatch rather than segmentation error alone. Within the main benchmark, MambaBot-UNet provided the strongest held-out HTS-Seg performance, while the complementary SegFormer analysis showed that transformer-based variants can further improve performance on the externally annotated DTS setting.

Together, the released datasets, model checkpoints, benchmark protocol, and web application provide a reproducible reference for spheroid segmentation under the documented bright-field imaging conditions. Within the limitations outlined above, SpheroSeg lowers the technical barrier for automated spheroid analysis and provides a practical foundation for future studies on annotation harmonization, domain adaptation, and downstream biological validation.

CRediT authorship contribution statement

Michal Průšek: Writing – review & editing, Writing – original draft, Visualization, Validation, Software, Methodology, Investigation, Formal analysis, Data curation, Conceptualization. **Adam Novozámský:** Writing – review & editing, Writing – original draft, Visualization, Validation, Supervision, Software, Resources, Project administration, Methodology, Funding acquisition, Data curation, Conceptualization. **Jan Škubník:** Writing – review & editing, Validation, Investigation, Data curation. **Vladimíra Svobodová Pavlíčková:** Writing – review & editing, Validation, Methodology, Investigation, Data curation. **Silvie Rimpelová:** Writing – review & editing, Writing – original draft, Validation, Supervision, Resources, Project administration, Methodology, Investigation, Funding acquisition, Data curation, Conceptualization.

Software and data availability

Software. SpheroSeg is available as a hosted web service at https://spherosegapp.utia.cas.cz/. For on-premises deployment, we provide a Dockerized version of the application, which provides the same core functionality while allowing local processing and data privacy control. The source code, installation instructions, and configuration files are hosted on GitHub at https://github.com/michalprusek/SpheroSeg under the MIT license.

Dataset and Model Weights. To support reproducibility of the reported benchmarks, we provide public access to the training data and model checkpoints. The SpheroHQ dataset (22,683 expert-corrected images) and the SpheroMix dataset (32,367 images from multiple sources) are available for download at https://staff.utia.cas.cz/novozada/spheroseg/. Model weights for the evaluated deep-learning configurations, including both pretrained and finetuned variants, are provided at the same location. These resources allow researchers to reproduce the reported benchmarks or apply the released models to their own bright-field spheroid segmentation data, subject to the dataset and domain limitations discussed in Section 4.1.

Citation and Usage. All resources are provided under an open-source license. Users of the datasets, models, or software should cite this article. When using SpheroMix, the original publications of the incorporated external datasets should also be cited accordingly.

Ethics statement

The study did not involve human participants or animals and therefore did not require ethical approval.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgments

The authors have no additional acknowledgments.

Appendix

A.1. Finetuning classic segmentation methods

Optimal parameter settings for segmentation methods differ greatly across projects and cell lines, meaning no single configuration consistently provides reliable results for all datasets. To optimize segmentation performance across different cell lines (e.g., PC-3, BxPC-3, A549, and MIA PaCa-2) and imaging conditions, we employed a hybrid optimization strategy that combines gradient-based and discrete search techniques. The objective was to maximize the mean IoU score between automatically generated segmentation masks and a small set of expert-annotated ground truth masks (GT). Specifically, continuous parameters (such as the adaptive thresholding window size or the weighting constants in the Niblack method) were optimized using the Adam optimizer [46], which follows the gradient of the IoU score estimated by numerical approximation. Discrete parameters, such as the size of morphological structuring elements or area thresholds, were tuned using a hill-climbing strategy that evaluates neighboring candidate values and selects the one that improves the overall score. The overall optimization problem was defined as:

$$\underset{\theta}{\text{maximize}} \quad \frac{1}{N}\sum_{i=1}^{N} \text{IoU}(M_i^{\text{pred}}(\theta), M_i^{\text{GT}}) \tag{1}$$

where N is the number of validation images, $M_i^{\text{pred}}(\theta)$ is the predicted segmentation mask for the ith image, and M_i^{GT} is the corresponding ground truth mask. θ denotes the set of tunable parameters, including the window size used in adaptive thresholding, the parameter in the Niblack method, the Gaussian blur standard deviation in the Canny edge detector, the standard deviation multiplier for hysteresis thresholding in edge detection, the dilation kernel size used in morphological postprocessing, and the minimum spheroid area threshold used to filter out small objects. The IoU score is computed as:

$$\text{IoU}(A(\theta), B) = \frac{|A(\theta) \cap B|}{|A(\theta) \cup B|} \tag{2}$$

where A and B denote the result of our method and GT, respectively.

The parameter vector θ thus comprised several components affecting different stages of the segmentation pipeline within post-processing steps. The entire process, along with the parameter optimization framework, was implemented in *prusek-spheroid*, a Python-based software tool with a graphical user interface [47], available on PyPI[1] and GitHub.[2] This tool was developed and used by the authors to perform batch processing of spheroid images and to generate the large-scale annotated dataset used for training deep learning models.

A.2. Dataset preparation for deep learning

We used two dataset configurations with binary semantic segmentation masks, where spheroid pixels are labeled as foreground (value 255) and all remaining pixels, including culture medium, debris, and imaging artifacts, are labeled as background (value 0). This formulation enables direct training of foreground–background segmentation models and supports downstream morphometric analysis. The combined dataset collection contains 32,367 images in SpheroMix, while the held-out HTS-Seg dataset is used only for independent out-of-distribution evaluation and is not included in the SpheroMix total.

[1] https://pypi.org/project/prusek-spheroid/
[2] https://github.com/michalprusek/prusek-spheroid/

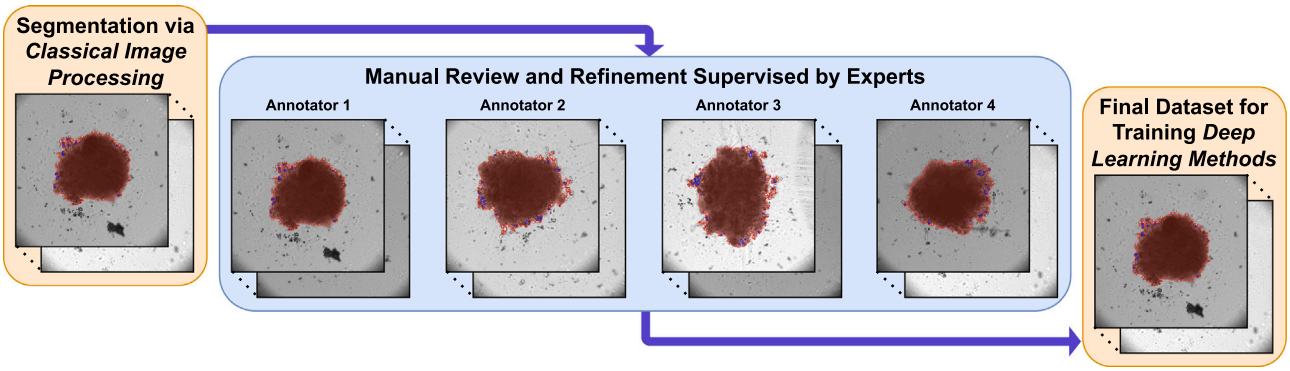

Fig. 5. Annotation process for accurate object boundary correction.

SpheroHQ (22,683 images): SpheroHQ is an expert-corrected bright-field spheroid dataset covering seven cancer cell lines: A549, BxPC-3, HeLa, MCF-7, MIA PaCa-2, PC-3, and UFH-001. The annotation workflow combines algorithmic pre-segmentation using the *prusek-spheroid* tool [47] with expert refinement in CVAT [48]. During expert review, artifacts such as microscope scratches and biological fibers were corrected or removed when they interfered with the spheroid mask; the annotation workflow is illustrated in Fig. 5.

SpheroMix (32,367 images): SpheroMix extends SpheroHQ by augmenting the training split with external manually annotated spheroid datasets (see Fig. 6):

- all 22,683 SpheroHQ images with semi-automatic expert-corrected annotations, while preserving the original SpheroHQ train/validation/test splits;
- *SLiMIA* [49]: 7862 unique images added to the training split;
- *DTS*: 1069 training images from the Deep-Tumour-Spheroid dataset [23] added to the training split, while preserving the original DTS train/test separation;
- *SpheroidJ*: 387 images from the SpheroidJ dataset [15] added to the training split.

The DTS test split of 366 images was never used during training. In this study, it is used only as an externally annotated cross-dataset evaluation set with a different annotation policy.

Critical Annotation Differences. External datasets employ annotation policies that differ from the SpheroHQ protocol. In SpheroHQ, annotations were drawn inclusively to delineate the full visible spheroid structure, including degrading peripheral regions, loosely attached cells, and diffuse boundaries when these were judged by experts to belong to the spheroid. In contrast, the DTS annotations follow a more conservative core-only policy, focusing primarily on the dense spheroid body and excluding peripheral or weakly contrasted regions.

This difference affects cross-dataset evaluation because a model trained on inclusive annotations may be penalized against a conservative ground truth even when it segments biologically plausible spheroid material. We therefore do not aggregate SpheroHQ and DTS results into a single unqualified performance claim. Instead, we report them separately throughout the evaluation: SpheroHQ measures within-distribution performance under the inclusive annotation policy, whereas DTS measures performance against an external annotation policy. The held-out HTS-Seg dataset is reported separately as an out-of-distribution benchmark with unseen acquisition conditions, cell lines, and annotation team.

Cell-Line-Based Data Splitting. We used strict cell-line-based splitting for SpheroHQ, where each cell line appears exclusively in one split: training, validation, or test. Cell lines were assigned to splits randomly rather than based on visual similarity or morphological characteristics. This reduces cell-line-level leakage and supports evaluation on cell phenotypes not seen during training (see Tables 3 and 4).

Table 3
Dataset distribution across splits.

Split	SpheroMix	SpheroHQ
Training	28,809[b]	19,491
Validation	2539	2539
Test	1,019[a]	653
Total	32,367	22,683

HTS-Seg (n = 96) is not part of SpheroMix and is reported separately as a held-out out-of-distribution benchmark.

[a] Includes 653 BxPC-3 images from the SpheroHQ test split and 366 images from the external DTS test split. The DTS test images are used only for evaluation and are not included in training.

[b] Contains 877 duplicate image pairs (3.04%), mainly within single experiments and across DTS/SpheroidJ datasets. Duplicates were detected via MD5 hashing but retained to preserve dataset integrity. Removing them yields 27,932 unique training images.

Table 4
Cell-line allocation across splits with strict separation.

Split	Cell lines
Training	
SpheroHQ core:	MCF-7 (7642), UFH-001 (6321), PC-3 (3292), MIA PaCa-2 (2236)
SpheroMix added:	51 cell lines from SLiMIA, 7 cell lines from SpheroidJ, DTS training cell lines
Validation	A549 (1689), HeLa (850)
Test	BxPC-3 (653)
	DTS (366; external annotation policy)

SpheroMix augments the training split with 9318 additional images from external sources (SLiMIA: 7862; DTS training: 1069; SpheroidJ: 387), providing broader imaging and morphological variability beyond the seven SpheroHQ cell lines. Importantly, only the DTS training split was used for model training; the DTS test split was strictly reserved for evaluation. For testing, SpheroHQ uses only BxPC-3 spheroids (653 images) for within-distribution evaluation, while the DTS test set (366 images) is used as an externally annotated cross-dataset evaluation set with a different core-only annotation policy. HTS-Seg is evaluated separately as a held-out out-of-distribution dataset.

External Held-Out Benchmark (HTS-Seg). To provide an independent out-of-distribution test set outside the SpheroMix training corpus, we additionally evaluated each model on the HTS-Seg dataset [22]. The public release we obtained contains 97 source bright-field images and 96 matching binary annotation masks; one image without a paired annotation was excluded. The remaining 96 image–mask pairs were converted to a 1024 × 1024 PNG test split, using Lanczos resampling for images and nearest-neighbor interpolation for binary masks.

HTS-Seg differs from both SpheroHQ and DTS along multiple axes: it uses a different acquisition platform, includes heterotypic spheroids with drug-induced morphological variability, contains cell lines not

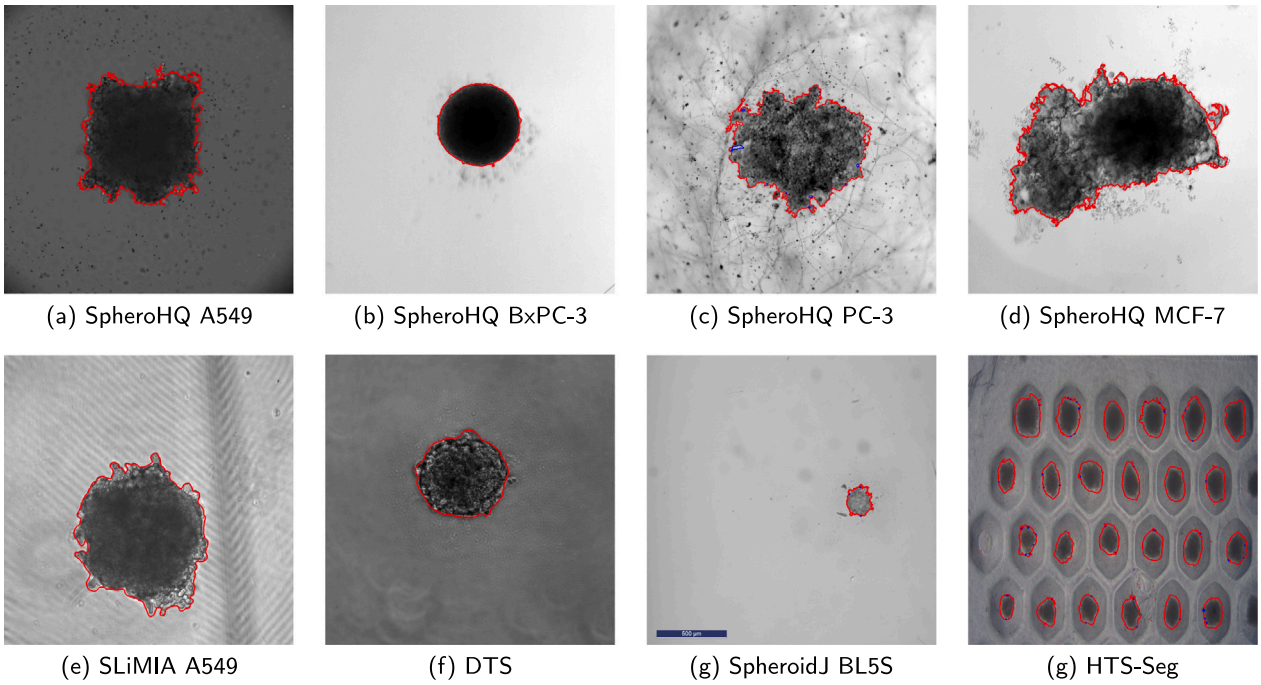

Fig. 6. Representative samples from the dataset collection with ground-truth mask overlays. Red contours indicate external spheroid boundaries, while blue contours indicate internal cavities or holes, both derived from the corresponding ground-truth annotations. Top row (a–d): SpheroHQ samples from four cell lines with expert-corrected annotations. Bottom row (e–g): external datasets used in this study, including SLiMIA, DTS, and SpheroidJ as sources incorporated into SpheroMix, and HTS-Seg as a held-out out-of-distribution benchmark. The examples illustrate differences in imaging conditions, spheroid morphology, object scale, and annotation protocols across data sources.

Table 5
Spheroid diameter statistics in pixels, computed per mask.[a].

Dataset	Spheroids analyzed	Min	Q1	Median	Q3	Max
SpheroHQ	36,135	6	12	255	446	1056
SpheroMix	47,467	4	19	268	449	1056

[a] The number of analyzed spheroids exceeds the number of images because extended cultivation or treatment can lead to spheroid fragmentation, with multiple segmented objects in a single image.

seen during pretraining or finetuning, and was annotated by an independent team. No HTS-Seg image was used during training or validation. We therefore report HTS-Seg separately as a held-out out-of-distribution benchmark in Table 2.

Dataset Standardization and Format. All images and masks were standardized to an 8-bit PNG format with a resolution of 1024 × 1024 pixels to ensure consistency across data sources. Original images spanning various formats (BMP, TIFF, JPEG) and bit depths (8-bit and 16-bit) were converted using adaptive normalization; 16-bit microscopy images underwent percentile-based scaling (2nd–98th percentile) to preserve contrast while limiting saturation. Binary segmentation masks contain two values: 0 for background and 255 for spheroid foreground, stored as single-channel PNG files. The final dataset organization uses separate `images/` and `masks/` directories for each split, with corresponding image–mask pairs sharing identical filenames.

Image Characteristics and Dataset Properties. Table 5 summarizes the spheroid diameter distribution. The larger number of analyzed spheroids relative to the number of images reflects cases in which spheroids fragment into multiple objects. Compared with SpheroHQ, SpheroMix contains a broader range of image sources and acquisition settings due to the incorporation of external datasets. This heterogeneity is reflected in the wider range of source resolutions and in the higher intensity standard deviation observed in SpheroMix (0.163) compared with SpheroHQ (0.148), while the normalized mean intensities remain comparable (0.642 for SpheroMix and 0.686 for SpheroHQ).

A.3. Training strategy

We used a two-stage training strategy designed to combine the diversity of SpheroMix with the higher annotation consistency of SpheroHQ. The first stage exposes the models to heterogeneous spheroid morphologies and imaging conditions, whereas the second stage adapts the decision boundaries to the expert-corrected SpheroHQ annotation policy.

Stage 1: Pretraining on Diverse Data. Models were initially pretrained on SpheroMix (32,367 images), which combines SpheroHQ with external spheroid datasets. This stage used the AdamW optimizer [50] with weight decay 10^{-4}, gradient clipping at L2 norm 1.0, and a OneCycleLR schedule [51] with a peak learning rate of 2×10^{-4}. The effective batch size was fixed at 16 samples per optimizer step for all architectures using gradient accumulation. Training was run for a maximum of 50 epochs with early stopping on validation IoU using patience 10 and a minimum improvement threshold of 10^{-4}.

Stage 2: Finetuning on Expert-Corrected Annotations. The pretrained models were subsequently finetuned on SpheroHQ (22,683 images), which contains expert-corrected annotations following a consistent inclusive annotation policy. Finetuning used the same optimizer, weight decay, gradient clipping, effective batch size, epoch budget, and early-stopping criterion as pretraining, but with a reduced OneCycleLR peak learning rate of 1×10^{-5}. No encoder freezing was used. This stage was intended to refine the models under the SpheroHQ

Table 6
Per-architecture training outcomes under the unified protocol. All configurations share AdamW with weight decay 10^{-4}, OneCycleLR with peak learning rate 2×10^{-4} for pretraining and 1×10^{-5} for finetuning, effective batch size 16, focal/Dice/IoU loss weights 1.0/1.0/0.5, gradient clipping at L2 norm 1.0, maximum 50 epochs, and early-stopping patience 10 on validation IoU. Per-GPU batch size and gradient accumulation are adjusted only to realize the constant effective batch size within available GPU memory.

Model	Stage	Params (M)	Per-GPU BS	Grad. accum.	Best epoch	Early-stop epoch	Best val. IoU
U-Net	Pretrain	45.20	4	4	18	28	0.9121
U-Net	Finetune	45.20	4	4	7	17	0.9145
HRNet	Pretrain	65.85	8	2	7	17	0.9028
HRNet	Finetune	65.85	4	4	36	46	0.9219
PSPNet	Pretrain	70.40	4	4	15	25	0.8902
PSPNet	Finetune	70.40	4	4	16	26	0.9070
CBAM-ResUNet	Pretrain	52.07	4	4	11	21	0.9088
CBAM-ResUNet	Finetune	52.07	2	8	17	27	0.9089
Our architectures:							
MA-ResUNet	Pretrain	19.57	4	4	13	23	0.9091
MA-ResUNet	Finetune	19.57	4	4	14	24	0.9108
MA-ResUNet-Mini	Pretrain	7.66	2	8	1	11	0.8971
MA-ResUNet-Mini	Finetune	7.66	2	8	29	39	0.9001
LC-ResUNet	Pretrain	58.06	4	4	19	29	0.9100
LC-ResUNet	Finetune	58.06	4	4	39	49	0.9199
MambaBot-UNet	Pretrain	90.75	4	4	29	39	0.9127
MambaBot-UNet	Finetune	90.75	4	4	24	34	0.9181

BS – batch size; Grad. accum. – gradient-accumulation steps. The effective batch size is 16 for all rows. Best val. IoU denotes the validation-set IoU at the selected checkpoint; test-set IoU is reported separately in Table 2.

annotation policy; its effect is therefore interpreted through the stratified evaluation in Table 2, rather than through a single aggregate pretrained-versus-finetuned score.

A.4. Loss function design

We used a multi-component loss function combining three complementary objectives:

$$\mathcal{L}_{\text{total}} = \alpha_1 \mathcal{L}_{\text{focal}} + \alpha_2 \mathcal{L}_{\text{dice}} + \alpha_3 \mathcal{L}_{\text{IoU}} \tag{3}$$

where $\alpha_1 = 1.0$, $\alpha_2 = 1.0$, and $\alpha_3 = 0.5$ balance the contributions of the individual components. The focal loss [52] addresses the class imbalance inherent in spheroid images, where background pixels typically constitute 80%–90% of the image:

$$\mathcal{L}_{\text{focal}} = -\alpha_{\text{focal}}(1 - p_t)^{\gamma} \log(p_t) \tag{4}$$

with $\alpha_{\text{focal}} = 0.8$ and $\gamma = 2$, where p_t represents the model's estimated probability for the ground-truth class. The modulating factor $(1-p_t)^{\gamma}$ reduces the contribution of well-classified examples and focuses training on harder pixels. The Dice loss directly optimizes foreground–background overlap and provides stable gradients for small objects:

$$\mathcal{L}_{\text{dice}} = 1 - \frac{2 \sum_i p_i g_i + \epsilon}{\sum_i p_i + \sum_i g_i + \epsilon} \tag{5}$$

where p_i and g_i denote predicted and ground-truth values for pixel i, with $\epsilon = 10^{-6}$ for numerical stability. The IoU loss directly optimizes the primary evaluation metric:

$$\mathcal{L}_{\text{IoU}} = 1 - \frac{\sum_i p_i g_i + \epsilon}{\sum_i p_i + \sum_i g_i - \sum_i p_i g_i + \epsilon} \tag{6}$$

This combination addresses class imbalance while optimizing overlap-based segmentation objectives.

A.5. Implementation details

Data Augmentation. We applied data augmentation using the Albumentations library [53] to improve robustness to common microscopy variability. The pipeline included geometric transformations (rotation up to ±45°, horizontal and vertical flips with $p = 0.5$, and elastic transformation with $\alpha = 120$ and $\sigma = 6$), photometric adjustments (brightness and contrast variation of ±20%, and CLAHE [54] with clip limit 2.0), and noise injection using Gaussian noise with variance 10–50. Grid and optical distortion were additionally applied with $p = 0.3$ to simulate microscopy-related geometric distortions.

Optimization and Training. Training was implemented in PyTorch [55] using mixed-precision computation with automatic loss scaling. All eight architectures in the main benchmark followed the standardized protocol described in Appendix A.3. To make the optimization conditions comparable across architectures, the effective batch size was held constant at 16 samples per optimizer step. The per-GPU mini-batch size and gradient-accumulation factor were adjusted only to satisfy memory constraints of individual architectures.

All runs used AdamW with weight decay 10^{-4}, gradient clipping at L2 norm 1.0, and OneCycleLR scheduling with peak learning rate 2×10^{-4} for pretraining and 1×10^{-5} for finetuning. Each stage used a maximum budget of 50 epochs and early stopping with patience 10 on validation IoU. No encoder freezing was applied in either stage.

Training Configurations and Outcomes. Table 6 reports the per-architecture training outcomes under the unified protocol. All hyperparameters that define the training recipe are shared across models; the table therefore reports only architecture-dependent quantities and memory-driven implementation details, namely parameter count, per-GPU batch size, gradient accumulation, best validation epoch, early-stop epoch, and best validation IoU.

All architectures reached their best validation checkpoint within the 50-epoch budget. The variation in per-GPU batch size is a memory-driven implementation detail, not a change in the effective optimization batch size, because gradients were accumulated to maintain 16 samples per optimizer step in every run. The test-set performance corresponding to these checkpoints is reported separately in Table 2.

A.6. Statistical evaluation methodology

The statistical analysis followed the stratified evaluation design summarized in Table 2. For each model and test setting, segmentation performance was evaluated using per-image Intersection over Union (IoU), which served as the primary ranking metric. IoU was selected because it penalizes both under-segmentation and over-segmentation and is sensitive to boundary disagreement between predicted and reference masks.

Uncertainty was quantified by non-parametric bootstrap resampling [56]. For each model and test setting, we resampled the per-image

Table 7
Inter-annotator agreement and annotation-policy gap in the 150-image DTS re-annotation pilot. Four annotators independently re-annotated the same DTS test images under the SpheroHQ inclusive policy, starting from identical Sauvola pre-annotations. Bootstrap 95% confidence intervals were computed from 10,000 resamples of per-image scores.

Metric	Value [95% CI]
Mean pairwise IoU across the 4 annotators	0.9399 [0.935–0.944]
Pixel-level Fleiss' κ across the 4 annotators	0.9629 [0.955–0.969]
STAPLE consensus IoU vs. each annotator	0.9643 [0.959–0.969]
STAPLE consensus IoU vs. original DTS GT	0.9201 [0.904–0.933]
Mean single-annotator IoU vs. original DTS GT	0.9105 [0.894–0.924]
Paired Wilcoxon test, within-policy vs. original DTS GT	$p = 5.4 \times 10^{-4}$

IoU values with replacement for 10,000 iterations and report the mean IoU together with the corresponding bootstrap 95% confidence interval. This procedure was applied independently to each test setting to avoid pooling images across different annotation policies or acquisition conditions.

Effect of Pretraining and Finetuning. The stratified evaluation changes the interpretation of the pretrained–finetuned comparison. Rather than reporting a single aggregate pretrained-versus-finetuned ranking across all test images, Table 2 presents the pretrained and finetuned variants side by side for each test setting.

Finetuning on the high-quality SpheroHQ annotations can improve or preserve within-distribution performance, but it does not consistently improve cross-dataset performance. In several architectures, finetuning reduces performance on the externally annotated DTS test split and on the held-out HTS-Seg benchmark. This indicates a specialization–generalization trade-off: the second training stage adapts the model more closely to the inclusive SpheroHQ annotation policy, but this specialization can reduce agreement with external annotation policies and unseen acquisition conditions.

Cross-Dataset Performance Analysis. Table 2 reports model performance separately for the three evaluation settings rather than as a single aggregate score. This stratification was used to distinguish within-distribution performance on SpheroHQ, evaluation under an external annotation policy on DTS, and broader out-of-distribution behavior on HTS-Seg.

The resulting changes in model ranking across the three test settings support this separated interpretation. High performance on the within-distribution SpheroHQ split does not necessarily translate to higher agreement with the externally annotated DTS masks or to better performance on the held-out HTS-Seg benchmark. The DTS results primarily reflect evaluation under a different annotation policy, whereas HTS-Seg represents a broader shift involving acquisition conditions, cell lines, and an independent annotation team. These settings are therefore interpreted separately throughout the manuscript.

Annotation Strategy Divergence. Performance differences between datasets are affected not only by model accuracy but also by divergent annotation strategies. The SpheroHQ dataset adopts an inclusive annotation policy that captures the visible spheroid structure, including peripheral degrading regions and loosely attached cells when these are judged by experts to belong to the spheroid. In contrast, the DTS annotations follow a more conservative core-only policy, focusing primarily on the dense spheroid body.

A model trained or finetuned on inclusive annotations may therefore be penalized when evaluated against the original DTS ground truth, even when it segments visible spheroid material consistently with the SpheroHQ policy. For this reason, SpheroHQ and DTS results are not merged into a single performance score. The completed DTS re-annotation pilot reported in Appendix A.7 further quantifies this annotation-policy gap directly.

A.7. DTS Re-Annotation pilot

To quantify the annotation-policy gap directly, we conducted a controlled re-annotation pilot on a 150-image subset of the DTS test partition. Four independent annotators, trained on the SpheroHQ inclusive annotation policy and blinded to the original DTS labels, re-annotated the same images. All annotators started from identical Sauvola pre-annotations generated by the same automatic pipeline used for the SpheroHQ pre-annotation stage. This design keeps the image content, pre-annotation method, annotation software, and annotation instructions fixed, while changing the annotation policy from the original DTS core-only convention to the SpheroHQ inclusive convention.

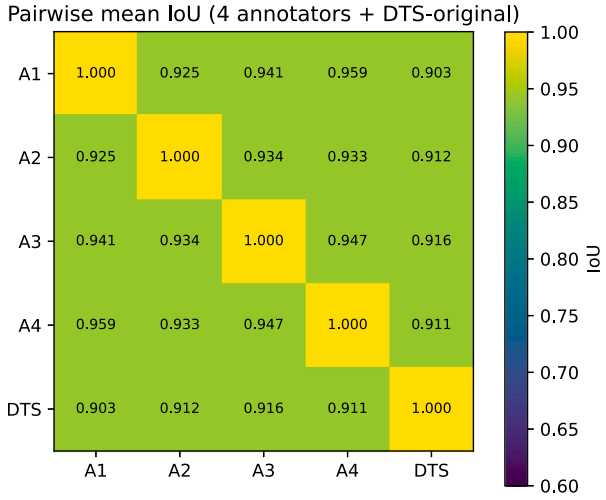

Fig. 7. Pairwise mean IoU across the four SpheroHQ-policy annotators (A1–A4) and the original DTS ground truth in the 150-image re-annotation pilot.

The annotators showed high agreement under the inclusive policy (Table 7). The mean pairwise IoU across the six annotator pairs was 0.9399 [0.935–0.944], and the pixel-level Fleiss' κ [57] across the four annotators was 0.9629 [0.955–0.969]. A STAPLE consensus mask computed from the four annotations was used as the within-policy reference for comparison with the original DTS ground truth.

The STAPLE consensus achieved a mean IoU of 0.9201 [0.904–0.933] against the original DTS core-only masks. This value is lower than the within-policy pairwise agreement, and the difference was statistically significant in a paired Wilcoxon test on per-image scores ($p = 5.4 \times 10^{-4}$, $n = 150$). However, the magnitude of the policy gap was moderate: the residual disagreement increased from 0.0601 within the inclusive policy to 0.0799 between the inclusive consensus and the original DTS labels, corresponding to a fold ratio of 1.33.

The pairwise heatmap in Fig. 7 illustrates this pattern. Agreement among the four inclusive-policy annotators is consistently high, ranging from 0.925 to 0.959. Agreement with the original DTS masks is uniformly lower, ranging from 0.903 to 0.916. This supports the interpretation that the observed disagreement reflects a systematic annotation-policy offset rather than an annotator-specific outlier.

The IoU of 0.9201 provides an empirical cross-policy agreement reference for interpreting the DTS results in Table 2. It quantifies the agreement between the SpheroHQ-style inclusive STAPLE consensus

Table 8
SegFormer transformer baseline under the stratified evaluation protocol. SegFormer variants were evaluated after SpheroMix training and after subsequent SpheroHQ finetuning. IoU values are reported as mean with bootstrap 95% confidence intervals computed from 10,000 resamples of per-image scores.

Model	Stage	SpheroHQ IoU [95% CI] (n = 653)	DTS IoU [95% CI] (n = 366)	HTS-Seg IoU [95% CI] (n = 96)	Time (ms)	Params (M)
SegFormer-B0	Pretr.	0.9272 [.919,.935]	0.9449 [.937,.952]	0.5139 [.488,.539]	**12.6**	3.71
SegFormer-B0	Finet.	**0.9280** [**.920,.936**]	0.9451 [.937,.952]	0.4931 [.467,.519]	**12.6**	3.71
SegFormer-B2	Pretr.	0.9232 [.913,.933]	0.9139 [.894,.932]	0.2493 [.226,.275]	30.4	27.35
SegFormer-B2	Finet.	0.9227 [.912,.932]	0.9131 [.893,.931]	0.2200 [.198,.244]	30.4	27.35
SegFormer-B4	Pretr.	0.8998 [.891,.909]	0.9484 [.941,.955]	0.5413 [.524,.559]	52.7	63.99
SegFormer-B4	Finet.	0.9255 [.917,.933]	**0.9493** [**.942,.955**]	**0.5655** [**.550,.581**]	52.7	63.99

Pretr. denotes training on the SpheroMix training split only. Finet. denotes subsequent finetuning on SpheroHQ. All checkpoints were evaluated using the same stratified test settings as Table 2.

and the original DTS core-only labels on the 150-image pilot subset, rather than defining an upper bound on model performance. Several model configurations exceed this value. As the benchmark covers the full 366-image DTS test split, the pilot indicates that part of the observed SpheroHQ–DTS gap reflects annotation-policy mismatch rather than segmentation error alone.

A.8. SegFormer transformer baseline

To complement the main eight-architecture benchmark in Table 2, we evaluated SegFormer [26] as a transformer-based segmentation baseline. SegFormer differs from the main benchmark by using a transformer encoder architecture and publicly available ADE-pretrained backbones. The results are therefore reported as a complementary analysis under the same stratified test protocol.

We evaluated SegFormer-B0, SegFormer-B2, and SegFormer-B4 in both SpheroMix-pretrained and SpheroHQ-finetuned configurations. The training setup followed the main protocol where possible, including 1024 × 1024 input resolution, effective batch size 16, OneCycleLR scheduling, mixed-precision training, and best-checkpoint selection based on validation IoU. The peak learning rate was 2×10^{-4} for SegFormer-B0 and SegFormer-B2, and 1×10^{-4} for SegFormer-B4 because of its larger model size.

The SegFormer results show that transformer-based models follow a cross-dataset profile that differs from the main benchmark. SegFormer-B0 and SegFormer-B4 achieved higher IoU on the externally annotated DTS test split than on the within-distribution SpheroHQ split, whereas SegFormer-B2 did not follow this pattern. This scale-dependent behavior supports the interpretation that the measured SpheroHQ–DTS gap depends not only on architecture family, but also on annotation-policy exposure, model capacity, and the degree of specialization introduced by second-stage SpheroHQ finetuning.

The effect of SpheroHQ finetuning was not uniform across SegFormer scales. For SegFormer-B4, finetuning improved both SpheroHQ performance and HTS-Seg performance, with HTS-Seg IoU increasing from 0.5413 [0.524–0.559] to 0.5655 [0.550–0.581]. In contrast, SegFormer-B0 changed only marginally on SpheroHQ and DTS but decreased on HTS-Seg, while SegFormer-B2 showed reduced performance after finetuning on both DTS and HTS-Seg. These results indicate that the benefit of second-stage finetuning is scale-dependent rather than uniform across transformer variants.

SegFormer-B4 after SpheroHQ finetuning achieved the strongest DTS and HTS-Seg performance within the complementary transformer analysis. Its DTS IoU exceeded all configurations in the main benchmark, whereas its HTS-Seg IoU remained below the MambaBot-UNet result in Table 2. Conversely, SegFormer-B0 provided the fastest transformer option, with an offline inference time of 12.6 ms per image and competitive performance on both DTS and HTS-Seg. We therefore interpret SegFormer as a complementary transformer baseline rather than as a replacement for the main benchmark: it demonstrates that transformer-based models can improve external-test performance in some settings, while the main benchmark remains necessary for comparing the architectures selected for the primary standardized evaluation. Among the SegFormer variants, SegFormer-B0 was selected for deployment because it provides the lowest transformer inference time.

A.9. Analysis of segmentation failures and dataset-specific challenges

Although the best models achieved high within-distribution performance, their errors are not uniformly distributed across all image types and annotation settings. We therefore examined representative failure cases from the SpheroHQ BxPC-3 and DTS test sets to identify limitations that are not fully captured by summary IoU values. The observed cases fall into three main categories.

Annotation-Policy Mismatch on DTS. A major source of apparent disagreement on DTS is the difference between the inclusive SpheroHQ annotation policy and the core-only DTS ground truth. In representative DTS cases, the deployed models may delineate the visible spheroid structure, including peripheral degrading regions, whereas the DTS reference mask contains only the denser core. These regions are therefore counted as over-segmentation during evaluation, even when they are consistent with the SpheroHQ annotation policy. This effect should be interpreted as an annotation-policy mismatch rather than as segmentation failure alone, as illustrated by the DTS example in Fig. 8. The DTS re-annotation pilot in Appendix A.7 quantifies this effect directly.

Boundary Ambiguity in Advanced Spheroids. Some advanced SpheroHQ spheroids exhibit gradual transitions between the dense core, peripheral tissue, and surrounding debris. In such cases, even under the inclusive SpheroHQ annotation policy, the exact boundary is ambiguous. The deployed models can shift the predicted contour inward or outward relative to the expert mask, leading to lower per-image IoU values than in compact early-stage spheroids. Representative examples are shown in Fig. 8. These cases illustrate that the main remaining difficulty is not the detection of the spheroid itself, but the consistent placement of boundaries in diffuse or degrading regions.

Artifact-Driven Misidentification. A third failure mode occurs when dark imaging artifacts resemble spheroid material. In such cases, a model may segment the artifact as part of the spheroid or as an additional object, particularly when the artifact has similar intensity and texture to the foreground. This limitation reflects the reliance of segmentation models on local intensity contrast and motivates human verification in ambiguous cases. In the SpheroSeg web application, such errors can be corrected by deleting or refining the corresponding polygon, but they remain relevant for fully automated batch processing without manual review. Representative examples are shown in Fig. 8.

A.10. SpheroSeg Application

Morphometric Parameter Definitions. The SpheroSeg application computes a standard set of morphometric parameters for each segmented spheroid polygon, including proper handling of internal holes such as necrotic cores. All measurements are computed in pixel units (px for lengths, px^2 for areas) and can be converted to physical units using the microscope calibration factor.

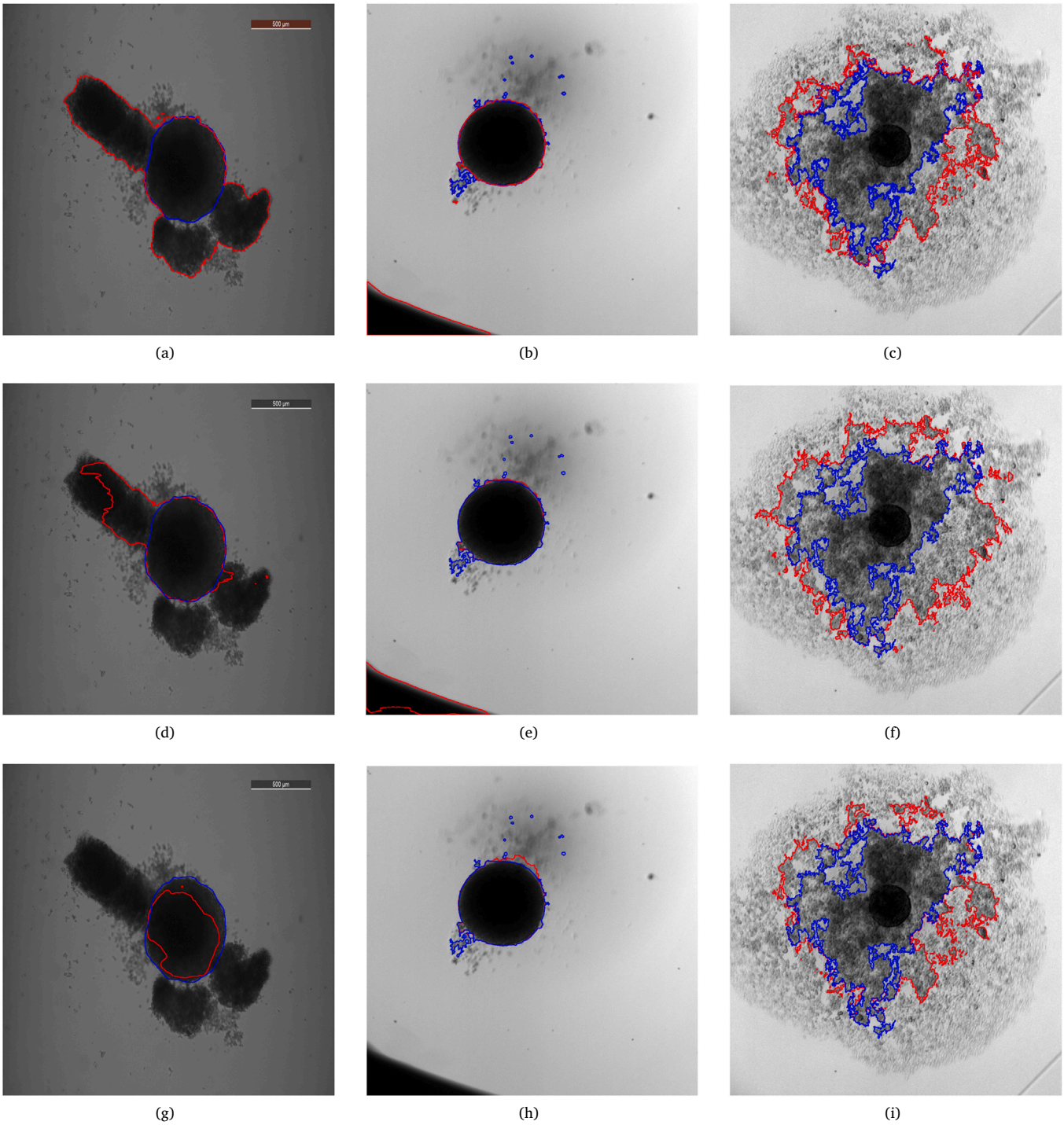

Fig. 8. Representative segmentation outputs from the three models deployed in the SpheroSeg web application. Blue contours denote ground-truth annotations and red contours denote model predictions. The rows show the deployed models: CBAM-ResUNet (top), MambaBot-UNet (middle), and SegFormer-B0 (bottom). The columns show the same representative cases across models: DTS example 280 (left), a SpheroHQ day-1 BxPC-3 image (middle), and a SpheroHQ day-4 BxPC-3 image (right). The examples illustrate differences between model outputs in practically relevant cases, including annotation-policy mismatch on DTS and ambiguous spheroid boundaries in SpheroHQ.

A.10.1. Area-based metrics

Area. The total enclosed area of the spheroid polygon, computed using the Shoelace formula (Gaussian area formula) with subtraction of internal hole areas:

$$A = \left| \frac{1}{2} \sum_{i=0}^{n-1} (x_i y_{i+1} - x_{i+1} y_i) \right| - \sum_{j=1}^{h} A_{\text{hole}_j} \tag{7}$$

where n is the number of vertices, indices are modular (vertex n connects to vertex 0), and h is the number of internal holes. This metric directly quantifies spheroid size and is fundamental for growth rate analysis.

Solidity. The ratio of the spheroid area to its convex hull area, measuring the presence of concavities:

$$S = \frac{A_{\text{spheroid}}}{A_{\text{convex hull}}} \tag{8}$$

Values range from 0 to 1, where 1 indicates a perfectly convex shape. Lower solidity values indicate irregular boundaries often associated with invasive phenotypes or drug-induced morphological changes.

Extent. The ratio of spheroid area to its axis-aligned bounding box (AABB) area:

$$E = \frac{A_{\text{spheroid}}}{\text{width} \times \text{height}} \tag{9}$$

where width $= \max(x) - \min(x)$ and height $= \max(y) - \min(y)$. This metric quantifies how efficiently the spheroid fills its axis-aligned bounding rectangle, with values approaching 1 for rectangular shapes and lower values for irregular or diagonal orientations. Note that this uses the axis-aligned bounding box (AABB), not the minimum area bounding rectangle which would be rotationally optimized.

A.10.2. Perimeter-based metrics

Perimeter. The total boundary length including both external and internal (hole) boundaries:

$$P = \sum_{i=0}^{n-1} \sqrt{(x_{i+1} - x_i)^2 + (y_{i+1} - y_i)^2} + \sum_{j=1}^{h} P_{\text{hole}_j} \tag{10}$$

Following the ImageJ/scikit-image convention, hole perimeters are added to the total, providing a complete measure of boundary complexity.

Circularity. A shape descriptor quantifying similarity to a perfect circle:

$$C = \frac{4\pi A}{P^2} \tag{11}$$

Theoretically bounded by 1 (perfect circle), this metric may slightly exceed 1 due to pixelization effects and is therefore clamped to [0, 1]. A square yields $C \approx 0.785$, providing a reference for interpreting intermediate values.

Convexity. The ratio of the convex hull perimeter to the actual perimeter:

$$\text{Conv} = \frac{P_{\text{convex hull}}}{P_{\text{spheroid}}} \tag{12}$$

This metric quantifies boundary smoothness, with values near 1 indicating smooth contours and lower values revealing irregular, potentially invasive edges.

Compactness. A normalized shape descriptor combining area and perimeter, defined as:

$$\text{Compactness} = \frac{P^2}{4\pi A} \tag{13}$$

This is the reciprocal of circularity, where values equal 1 for a perfect circle and increase for more complex shapes. This formulation follows the standard definition used in ImageJ and ensures scale invariance. Alternative definitions exist (e.g., $\sqrt{A}/P$), but we adopt the most common formulation for consistency with existing literature.

A.10.3. Size metrics

Equivalent Diameter. The diameter of a circle with the same area as the spheroid:

$$D_{\text{eq}} = \sqrt{\frac{4\,A}{\pi}} \tag{14}$$

This provides a single size measure facilitating comparison between spheroids of different shapes, particularly useful for growth curve analysis.

Feret Diameters. Caliper diameters measuring the spheroid extent in various orientations, computed using the rotating calipers algorithm [58] on the convex hull:

- **Maximum Feret Diameter:** The maximum distance between parallel tangent lines (longest caliper diameter)
- **Minimum Feret Diameter:** The minimum distance between parallel tangent lines (shortest caliper diameter)
- **Orthogonal Feret Diameter:** The width measured perpendicular to the maximum Feret axis

Aspect Ratio. The ratio of maximum to minimum Feret diameters:

$$AR = \frac{D_{\text{Feret max}}}{D_{\text{Feret min}}} \tag{15}$$

Values equal to 1 indicate circular shapes, while higher values quantify elongation, relevant for characterizing drug-induced morphological transitions.

Bounding Box Dimensions. The width and height of the axis-aligned bounding box (AABB) enclosing the spheroid, providing rapid size estimates without complex geometric calculations.

A.10.4. Implementation considerations

All metrics incorporate numerical safeguards including division-by-zero protection (returning 0 for undefined cases), NaN/Infinity checking, and physical bound clamping (e.g., circularity ≤ 1). The computational complexity is $O(n)$ for most metrics, where n is the vertex count, except for convex hull-based calculations requiring $O(n \log n)$ via Graham scan algorithm [59]. Polygons require a minimum of 3 vertices for valid computation, and all algorithms produce deterministic results ensuring reproducibility across analyses.

The metrics implementation follows established conventions from ImageJ [60] and scikit-image [61], ensuring compatibility with existing image analysis workflows in the field. For spheroids with internal necrotic cores, the hole subtraction in area calculation but addition in perimeter calculation correctly captures both size reduction and boundary complexity increase associated with central necrosis.

A.10.5. Models and performance

Three deployed models are provided in the SpheroSeg web application to cover complementary practical use cases. The selection was informed by the stratified benchmark in Table 2, the complementary SegFormer analysis in Table 8, and deployment-oriented considerations of speed, robustness, and annotation-policy behavior.

- **CBAM-ResUNet**: Included as the high-accuracy SpheroHQ option. The finetuned CBAM-ResUNet achieved the best within-distribution SpheroHQ IoU among the main benchmark configurations (0.9478 [0.944–0.952]), reflecting strong performance under the SpheroHQ inclusive annotation policy. Because its external-test performance is lower, especially on HTS-Seg, it should be interpreted primarily as the high-accuracy option for SpheroHQ-like images rather than as a universal model.
- **SegFormer-B0**: Included as the fast transformer-based option. In the complementary SegFormer analysis, the pretrained SegFormer-B0 provided strong DTS performance (0.9449 [0.937–0.952]) and the lowest offline inference time among the evaluated SegFormer variants (12.6 ms per image). It is therefore deployed for use cases where low latency is prioritized.

Table 9
End-to-end workflow performance breakdown for the three models deployed in the SpheroSeg web application. Values are reported in milliseconds as means with 95% confidence intervals from 522 real production measurements per model.

Step	CBAM-ResUNet	MambaBot-UNet	SegFormer-B0
Image Upload	5.55 ± 0.11	5.26 ± 0.10	**5.25 ± 0.09**
Queue Processing	9.26 ± 0.18	8.76 ± 0.17	**8.75 ± 0.15**
Preprocessing	**0.58 ± 0.01**	0.60 ± 0.01	**0.58 ± 0.01**
ML Inference	413.34 ± 0.87	235.62 ± 0.92	**171.63 ± 1.07**
Postprocessing	**0.14**	0.15	0.15
Database Write	1.85 ± 0.04	**1.75 ± 0.03**	**1.75 ± 0.03**
Thumbnail Generation	1.48 ± 0.03	**1.40 ± 0.03**	**1.40 ± 0.02**
WebSocket Notification	0.37	**0.35**	**0.35**
Total	432.6 ± 0.97	253.9 ± 1.03	**189.9 ± 1.16**

Table 10
Detailed production performance statistics for the three models deployed in the SpheroSeg web application. Values are based on 522 real production measurements per model.

Metric	CBAM-ResUNet	MambaBot-UNet	SegFormer-B0
Inference Mean (ms)	413.3	235.6	**171.6**
Inference 95% CI	[412.5, 414.2]	[234.7, 236.5]	[170.6, 172.7]
Median (ms)	416.2	238.5	**175.2**
P5–P95 (ms)	395.6–426.2	216.3–248.9	**150.8–184.3**
CV (%)	**2.4**	4.6	7.2
Total E2E Mean (ms)	432.6	253.9	**189.9**
Total E2E Median (ms)	434.8	255.8	**192.3**
E2E P5–P95 (ms)	413.2–449.1	233.0–269.5	**167.7–205.4**
E2E CV (%)	**2.6**	4.7	7.1

Table 11
Throughput metrics for the three models deployed in the SpheroSeg web application. Single-image throughput reflects individual end-to-end and inference-only processing, whereas batch throughput reports inference-only GPU throughput at the corresponding optimal batch size.

Metric	CBAM-ResUNet	MambaBot-UNet	SegFormer-B0
Single E2E Throughput (img/s)	2.3	3.9	**5.3**
Inference-only Throughput (img/s)	2.4	4.2	**5.8**
Batch Throughput (img/s)	5.0	10.9	**42.1**
Optimal Batch Size	2	2	**4**
Batch Speedup	2.18×	2.76×	**7.99**×

Single E2E throughput includes the complete production serving path. Inference-only throughput isolates the model inference stage for individual requests. Batch throughput measures inference-only batch processing and therefore excludes non-model overhead such as queue handling, preprocessing, postprocessing, and database operations.

- **MambaBot-UNet**: Included as the state-space option with the strongest external-test performance in the main benchmark. The finetuned MambaBot-UNet achieved the highest HTS-Seg IoU among the main benchmark configurations (0.5867 [0.563–0.607]) and also achieved the highest DTS IoU within the main benchmark (0.9351 [0.926–0.943]). This makes it suitable for cases where external-test robustness is prioritized.

Together, these models provide complementary deployment options rather than a single universally optimal configuration. Users working with SpheroHQ-like bright-field images may prefer CBAM-ResUNet for within-distribution accuracy, SegFormer-B0 for low-latency transformer inference, or MambaBot-UNet for strong external-test performance within the main benchmark. For images acquired under substantially different conditions, the stratified results in Tables 2 and 8 should be used as a guide, and manual verification or additional finetuning may be required.

End-to-End Performance Analysis. The web application provides sub-second response times for complete segmentation workflows across all three deployed models. Table 9 reports the detailed breakdown of processing times from image upload to the final segmentation result.

Performance Measurement Methodology. End-to-end performance measurements were conducted on the production server equipped with an NVIDIA RTX A5000 GPU (24 GB VRAM) using 522 real test images. The measurement script captured actual HTTP API calls to the deployed service, including image upload, task queue processing, preprocessing, GPU inference, postprocessing, database operations, thumbnail generation, and WebSocket notifications. Each component timing was extracted from server-side instrumentation. The ML service runs in a Docker container with CUDA 12.1, ensuring consistent deployment conditions across measurements. Unless stated otherwise, timings are reported as mean values with 95% confidence intervals calculated using Student's t-distribution.

Throughput and Resource Utilization. The application supports both single-image and batch processing modes. Table 9 reports the end-to-end workflow latency measured on the production deployment, Table 10 provides detailed latency statistics, and Table 11 summarizes the corresponding throughput. In routine operation, all deployed models achieved sub-second per-image latency. The total end-to-end latency ranged from 189.9 ms for SegFormer-B0 to 432.6 ms for CBAM-ResUNet, with MambaBot-UNet occupying an intermediate operating point at 253.9 ms.

The throughput measurements follow the same pattern. In single-image end-to-end mode, throughput ranged from 2.3 images/s for CBAM-ResUNet to 5.3 images/s for SegFormer-B0. Inference-only throughput for individual requests ranged from 2.4 images/s for CBAM-ResUNet to 5.8 images/s for SegFormer-B0. Under isolated batch inference, throughput increased to 5.0 images/s for CBAM-ResUNet, 10.9 images/s for MambaBot-UNet, and 42.1 images/s for SegFormer-B0 at the corresponding optimal batch sizes. These batch-throughput values should not be interpreted as full application throughput because they exclude non-model overhead such as queue handling, preprocessing, postprocessing, and database operations. Under the documented deployment conditions, the system supports both interactive segmentation and GPU-accelerated batch processing, with the final choice of model depending on the desired balance between segmentation behavior, latency, and external-test performance.

References

[1] M.T. Hoque, L.C.E. Windus, C.J. Lovitt, V.M. Avery, PCaAnalyser: A 2D-image analysis based module for effective determination of prostate cancer progression in 3D culture, PLOS ONE 8 (11) (2013) 1–13, http://dx.doi.org/10.1371/journal.pone.0079865.

[2] V. Härmä, H.-P. Schukov, A. Happonen, I. Ahonen, J. Virtanen, H. Siitari, M. Åkerfelt, J. Lötjönen, M. Nees, Quantification of dynamic morphological drug responses in 3D organotypic cell cultures by automated image analysis, PLoS ONE 9 (5) (2014) 1–18, http://dx.doi.org/10.1371/journal.pone.0096426.

[3] W. Chen, C. Wong, E. Vosburgh, A.J. Levine, D.J. Foran, E.Y. Xu, High-throughput image analysis of tumor spheroids: A user-friendly software application to measure the size of spheroids automatically and accurately, J. Vis. Exp. (89) (2014) e51639, http://dx.doi.org/10.3791/51639.

[4] D.P. Ivanov, T.L. Parker, D.A. Walker, C. Alexander, M.B. Ashford, P.R. Gellert, M.C. Garnett, Multiplexing spheroid volume, resazurin and acid phosphatase viability assays for high-throughput screening of tumour spheroids and stem cell neurospheres, PLoS ONE 9 (8) (2014) 1–14, http://dx.doi.org/10.1371/journal.pone.0103817.

[5] Y. Hou, J. Konen, D.J. Brat, A.I. Marcus, L.A.D. Cooper, TASI: A software tool for spatial-temporal quantification of tumor spheroid dynamics, Sci. Rep. 8 (1) (2018) 7248, http://dx.doi.org/10.1038/s41598-018-25337-4.

[6] I. Grexa, A. Diosdi, M. Harmati, A. Kriston, N. Moshkov, K. Buzas, V. Pietiäinen, K. Koos, P. Horvath, SpheroidPicker for automated 3D cell culture manipulation using deep learning, Sci. Rep. 11 (1) (2021) 14813, http://dx.doi.org/10.1038/s41598-021-94217-1.

[7] R. Barrow, J.N. Wilkinson, Y. He, M. Callaghan, A. Brüning-Richardson, M. Dunning, L.F. Stead, SpheroidAnalyseR-an online platform for analyzing data from 3D spheroids or organoids grown in 96-well plates, J. Biol. Methods 9 (4) (2022) e163, http://dx.doi.org/10.14440/jbm.2022.388.

[8] F. Piccinini, AnaSP: A software suite for automatic image analysis of multicellular spheroids, Comput. Methods Programs Biomed. 119 (1) (2015) 43–52, http://dx.doi.org/10.1016/j.cmpb.2015.02.006.

[9] C. Moriconi, V. Palmieri, R. Di Santo, G. Tornillo, M. Papi, G. Pilkington, M. De Spirito, M. Gumbleton, INSIDIA: A FIJI macro delivering high-throughput and high-content spheroid invasion analysis, Biotechnol. J. 12 (10) (2017) 1700140, http://dx.doi.org/10.1002/biot.201700140.

[10] G. Perini, E. Rosa, G. Friggeri, L. Di Pietro, M. Barba, O. Parolini, G. Ciasca, C. Moriconi, M. Papi, M. De Spirito, V. Palmieri, INSIDIA 2.0 high-throughput analysis of 3D cancer models: Multiparametric quantification of graphene quantum dots photothermal therapy for glioblastoma and pancreatic cancer, Int. J. Mol. Sci. 23 (6) (2022) http://dx.doi.org/10.3390/ijms23063217.

[11] M. Vinci, S. Gowan, F. Boxall, L. Patterson, M. Zimmermann, W. Court, C. Lomas, M. Mendiola, D. Hardisson, S.A. Eccles, Advances in establishment and analysis of three-dimensional tumor spheroid-based functional assays for target validation and drug evaluation, BMC Biol. 10 (1) (2012) 29, http://dx.doi.org/10.1186/1741-7007-10-29.

[12] L.R. Cisneros Castillo, A.-D. Oancea, C. Stüllein, A. Régnier-Vigouroux, A novel Computer-Assisted approach to evaluate multicellular tumor spheroid invasion assay, Sci. Rep. 6 (1) (2016) 35099, http://dx.doi.org/10.1038/srep35099.

[13] A.K. Mali, S. Murugappan, J.R. Prasad, S.A.M. Tofail, N.D. Thorat, A deep learning pipeline for morphological and viability assessment of 3D cancer cell spheroids, BioRxiv (2025) http://dx.doi.org/10.1101/2025.01.20.633939.

[14] F. Piccinini, A. Peirsman, M. Stellato, J.-c. Pyun, M.M. Tumedei, M. Tazzari, O. De Wever, A. Tesei, G. Martinelli, G. Castellani, Deep learning-based tool for morphotypic analysis of 3D multicellular spheroids, J. Mech. Med. Biol. 23 (06) (2023) 2340034, http://dx.doi.org/10.1142/S0219519423400341.

[15] D. Lacalle, H.A. Castro-Abril, T. Randelovic, C. Domínguez, J. Heras, E. Mata, G. Mata, Y. Méndez, V. Pascual, I. Ochoa, SpheroidJ: An open-source set of tools for spheroid segmentation, Comput. Methods Programs Biomed. 200 (2021) 105837, http://dx.doi.org/10.1016/j.cmpb.2020.105837.

[16] A. Akshay, M. Katoch, M. Abedi, N. Shekarchizadeh, M. Besic, F.C. Burkhard, A. Bigger-Allen, R.M. Adam, K. Monastyrskaya, A.H. Gheinani, SpheroScan: a user-friendly deep learning tool for spheroid image analysis, (ISSN: 2047-217X) 2023, http://dx.doi.org/10.1093/gigascience/giad082.

[17] S.K. Sadanandan, J. Karlsson, C. Wählby, Spheroid segmentation using multiscale deep adversarial networks, in: 2017 IEEE International Conference on Computer Vision Workshops, ICCVW, 2017, pp. 36–41, http://dx.doi.org/10.1109/ICCVW.2017.11.

[18] O. Ronneberger, P. Fischer, T. Brox, U-net: Convolutional networks for biomedical image segmentation, in: Medical Image Computing and Computer-Assisted Intervention (MICCAI 2015), LNCS 9351, in: Lecture Notes in Computer Science, vol. 9351, Springer, Cham, 2015, pp. 234–241, http://dx.doi.org/10.1007/978-3-319-24574-4_28.

[19] J. Long, E. Shelhamer, T. Darrell, Fully Convolutional Networks for Semantic Segmentation, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, CVPR, IEEE, 2015, pp. 3431–3440.

[20] K. He, G. Gkioxari, P. Dollár, R. Girshick, Mask R-CNN, in: Proceedings of the IEEE International Conference on Computer Vision, ICCV, IEEE, 2017, pp. 2980–2988.

[21] Z. Chen, N. Ma, X. Sun, Q. Li, Y. Zeng, F. Chen, S. Sun, J. Xu, J. Zhang, H. Ye, J. Ge, Z. Zhang, X. Cui, K. Leong, Y. Chen, Z. Gu, Automated evaluation of tumor spheroid behavior in 3D culture using deep learning-based recognition, Biomaterials 272 (2021) 120770, http://dx.doi.org/10.1016/j.biomaterials.2021.120770.

[22] A. Tahmasbi, A. Ahvaraki, E. Behroodi, A. Ghaffari, Z. Bagheri, F. Vakhshiteh, H. Latifi, Z. Madjd, A high-resolution dataset for AI-driven segmentation and analysis of drug-treated breast tumor spheroids, Comput. Methods Programs Biomed. 274 (2026) 109141, http://dx.doi.org/10.1016/j.cmpb.2025.109141.

[23] M. Streller, S. Michlíková, W. Ciecior, K. Lönnecke, L.A. Kunz-Schughart, S. Lange, A. Voss-Böhme, Image segmentation of treated and untreated tumor spheroids by fully convolutional networks, 2024, http://dx.doi.org/10.48550/arXiv.2405.01105, ArXiv.

[24] M. García-Domínguez, C. Domínguez, J. Heras, E. Mata, V. Pascual, Deep style transfer to deal with the domain shift problem on spheroid segmentation, Neurocomputing 569 (2024) 127105, http://dx.doi.org/10.1016/j.neucom.2023.127105.

[25] F. Çelebi, D. Boyvat, S. Ayaz-Guner, K. Tasdemir, K. Icoz, Improved senescent cell segmentation on bright-field microscopy images exploiting representation level contrastive learning, Int. J. Imaging Syst. Technol. 34 (2) (2024) e23052, http://dx.doi.org/10.1002/ima.23052.

[26] E. Xie, W. Wang, Z. Yu, A. Anandkumar, J.M. Alvarez, P. Luo, SegFormer: Simple and Efficient Design for Semantic Segmentation with Transformers, in: Advances in Neural Information Processing Systems (NeurIPS), vol. 34, 2021, pp. 12077–12090.

[27] A.L.V. Zumaya, V.S. Pavlíčková, S. Rimpelová, M. Štějdířová, M. Fulem, I. Křížová, P. Ulbrich, P. Řezanka, F. Hassouna, PLGA-based nanocarriers for combined delivery of colchicine and purpurin 18 in cancer therapy: Multimodal approach employing cancer cell spheroids, Int. J. Pharm. 657 (2024) 124170, http://dx.doi.org/10.1016/j.ijpharm.2024.124170.

[28] J. Friedrich, C. Seidel, R. Ebner, L.A. Kunz-Schughart, Spheroid-based drug screen: considerations and practical approach, Nat. Protoc. 4 (3) (2009) 309–324, http://dx.doi.org/10.1038/nprot.2008.226.

[29] J. Sauvola, M. Pietikäinen, Adaptive document image binarization, Pattern Recognit. 33 (2) (2000) 225–236, http://dx.doi.org/10.1016/S0031-3203(99)00055-2.

[30] W. Niblack, An Introduction to Digital Image Processing, Strandberg Publishing Company, DNK, 1985.

[31] N. Otsu, A threshold selection method from gray-level histograms, IEEE Trans. Syst. Man Cybern. 9 (1) (1979) 62–66.

[32] S. Suzuki, K. Abe, Topological structural analysis of digitized binary images by border following, Comput. Vis. Graph. Image Process. 30 (1) (1985) 32–46, http://dx.doi.org/10.1016/0734-189X(85)90016-7.

[33] J. Canny, A computational approach to edge detection, IEEE Trans. Pattern Anal. Mach. Intell. PAMI-8 (6) (1986) 679–698, http://dx.doi.org/10.1109/TPAMI.1986.4767851.

[34] K. Sun, Y. Zhao, B. Jiang, T. Cheng, B. Xiao, D. Liu, Y. Mu, X. Wang, W. Liu, J. Wang, High-resolution representations for labeling pixels and regions, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops, CVPRW, 2019, pp. 1–13, http://dx.doi.org/10.48550/arXiv.1904.04514.

[35] H. Zhao, J. Shi, X. Qi, X. Wang, J. Jia, Pyramid scene parsing network, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, CVPR, 2017, pp. 2881–2890, http://dx.doi.org/10.1109/CVPR.2017.660.

[36] S. Woo, J. Park, J.-Y. Lee, I.S. Kweon, CBAM: Convolutional block attention module, in: Proceedings of the European Conference on Computer Vision, ECCV, 2018, pp. 3–19, http://dx.doi.org/10.1007/978-3-030-01234-2_1.

[37] F.I. Diakogiannis, F. Waldner, P. Caccetta, C. Wu, ResUNet-a: a deep learning framework for semantic segmentation of remotely sensed data, ISPRS J. Photogramm. Remote Sens. 162 (2020) 94–114, http://dx.doi.org/10.1016/j.isprsjprs.2020.01.013.

[38] P. Yang, H. Li, K. Yu, J. Li, Y. Li, Simam: A simple, parameter-free attention module for convolutional neural networks, in: Proceedings of the 38th International Conference on Machine Learning, ICML, 2021, pp. 11863–11874.

[39] D. Misra, T. Nalamada, R. Arora, A. Ghosh, Rotate to attend: Convolutional triplet attention module, in: 2021 IEEE Winter Conference on Applications of Computer Vision, WACV, 2021, pp. 3139–3148, http://dx.doi.org/10.1109/WACV48630.2021.00318.

[40] A. Gu, T. Dao, Mamba: Linear-time sequence modeling with selective state spaces, 2023, http://dx.doi.org/10.48550/arXiv.2312.00752, arXiv:2312.00752.

[41] L. Zhu, B. Liao, Q. Zhang, X. Wang, W. Liu, X. Wang, Vision mamba: Efficient visual representation learning with bidirectional state space model, 2024, http://dx.doi.org/10.48550/arXiv.2401.09417, arXiv:2401.09417.

[42] W. Liao, Y. Zhu, X. Wang, C. Pan, Y. Wang, L. Ma, LightM-UNet: Mamba assists in lightweight unet for medical image segmentation, 2024, http://dx.doi.org/10.48550/arXiv.2403.05246, arXiv:2403.05246.

[43] J. Ma, F. Li, B. Wang, U-Mamba: Enhancing long-range dependency for biomedical image segmentation, 2024, http://dx.doi.org/10.48550/arXiv.2401.04722, arXiv:2401.04722.

[44] A. Archit, S. Nair, N. Khalid, P. Hilt, V. Rajashekar, M. Freitag, S. Gupta, A. Dengel, S. Ahmed, C. Pape, Segment anything for microscopy, BioRxiv (2023) http://dx.doi.org/10.1101/2023.08.21.554208.

[45] S.K. Warfield, K.H. Zou, W.M. Wells, Simultaneous truth and performance level estimation (STAPLE): An algorithm for the validation of image segmentation, IEEE Trans. Med. Imaging 23 (7) (2004) 903–921, http://dx.doi.org/10.1109/TMI.2004.828354.

[46] D.P. Kingma, J. Ba, Adam: A method for stochastic optimization, in: Proceedings of the 3rd International Conference on Learning Representations (ICLR 2015), 2015, http://dx.doi.org/10.48550/arXiv.1412.6980, ArXiv.

[47] M. Průšek, prusek-spheroid: A Python Tool for Automated Spheroid Segmentation and Analysis, 2024, URL https://github.com/michalprusek/prusek-spheroid, (Accessed 30 June 2025).

[48] CVAT.ai, CVAT: Computer vision annotation tool, 2020, (Accessed 30 June 2025), https://github.com/cvat-ai/cvat.

[49] E. Blondeel, A. Peirsman, S. Vermeulen, F. Piccinini, F. De Vuyst, D. Estêvão, S. Al-Jamei, M. Bedeschi, G. Castellani, T. Cruz, S. Dedeyne, M.J. Oliveira, S. Kawakita, H.T. Nguyen, L.A. Kunz-Schughart, S. Lee, N. Marino, P. Steigemann, S. Takayama, A. Tesei, N. Zablowsky, P. Blondeel, O. De Wever, The spheroid light microscopy image atlas for morphometrical analysis of three-dimensional cell cultures, Sci. Data 12 (1) (2025) 283, http://dx.doi.org/10.1038/s41597-025-04441-x.

[50] I. Loshchilov, F. Hutter, Decoupled weight decay regularization, in: International Conference on Learning Representations, ICLR, 2019, pp. 1–19, arXiv:1711.05101.

[51] L.N. Smith, A disciplined approach to neural network hyper-parameters: Part 1 – learning rate, batch size, momentum, and weight decay, 2018, http://dx.doi.org/10.48550/arXiv.1803.09820, ArXiv.

[52] T.-Y. Lin, P. Goyal, R. Girshick, K. He, P. Dollár, Focal loss for dense object detection, in: Proceedings of the IEEE International Conference on Computer Vision, ICCV, 2017, pp. 2980–2988, http://dx.doi.org/10.1109/ICCV.2017.324.

[53] A. Buslaev, V.I. Iglovikov, E. Khvedchenya, A. Parinov, M. Druzhinin, A.A. Kalinin, Albumentations: Fast and flexible image augmentations, Information 11 (2) (2020) http://dx.doi.org/10.3390/info11020125.

[54] K. Zuiderveld, Contrast limited adaptive histogram equalization, in: Graphics Gems IV, Academic Press Professional, Inc., USA, 1994, pp. 474–485.

[55] A. Paszke, S. Gross, F. Massa, A. Lerer, J. Bradbury, G. Chanan, T. Killeen, Z. Lin, N. Gimelshein, L. Antiga, A. Desmaison, A. Köpf, E. Yang, Z. DeVito, M. Raison, A. Tejani, S. Chilamkurthy, B. Steiner, L. Fang, J. Bai, S. Chintala, PyTorch: An imperative style, high-performance deep learning library, in: Advances in Neural Information Processing Systems (NeurIPS), vol. 32, 2019, pp. 8024–8035.

[56] B. Efron, Bootstrap methods: Another look at the Jackknife, Ann. Stat. 7 (1) (1979) 1–26, http://dx.doi.org/10.1214/aos/1176344552.

[57] J.L. Fleiss, Measuring nominal scale agreement among many raters, Psychol. Bull. 76 (5) (1971) 378–382, http://dx.doi.org/10.1037/h0031619.

[58] G.T. Toussaint, Solving geometric problems with the rotating calipers, in: Proc. IEEE MELECON, 1983, pp. A10/1–A10/4.

[59] R.L. Graham, An efficient algorithm for determining the convex hull of a finite planar set, Inform. Process. Lett. 1 (4) (1972) 132–133, http://dx.doi.org/10.1016/0020-0190(72)90045-2.

[60] C.T. Rueden, J. Schindelin, M.C. Hiner, T. DeZonia, A.E. Walter, E.T. Arena, K.W. Eliceiri, ImageJ2: ImageJ for the next generation of scientific image data, BMC Bioinformatics 18 (2017) 529.

[61] 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.