Deep Learning-Based Tool For Morphotypic Analysis Of 3D Multicellular Spheroids

(Article begins on next page)

22 February 2025

DEEP LEARNING-BASED TOOL FOR MORPHOTYPIC
ANALYSIS OF 3D MULTICELLULAR SPHEROIDS

FILIPPO PICCININI [1,2,*], ARNE PEIRSMAN [3,4,5], MARIACHIARA STELLATO [2], JAE-CHUL PYUN [6], MARIA M. TUMEDEI [1], MARCELLA TAZZARI [1], OLIVIER DE WEVER [3], ANNA TESEI [1], GIOVANNI MARTINELLI [1], and GASTONE CASTELLANI [2]

[1] IRCCS Istituto Romagnolo per lo Studio dei Tumori (IRST) ''Dino Amadori'',
Meldola (FC), Italy.
[2] Department of Medical and Surgical Sciences (DIMEC),
University of Bologna, Bologna, Italy.
[3] Laboratory of Experimental Cancer Research,
Cancer Research Institute, Ghent, Belgium.
[4] Department of Human Structure and Repair,
Ghent University, Ghent, Belgium.
[5] Plastic, Reconstructive and Aesthetic Surgery,
Ghent University Hospital, Ghent, Belgium.
[6] Department of Materials Science and Engineering,
Yonsei University, Seoul, South Korea.
* Corresponding Author: Filippo Piccinini (F.P.). Email: f.piccinini@unibo.it; filippo.piccinini@irst.emr.it

Introduction: 3D multicellular spheroids are fundamental *in vitro* tools for studying *in vivo* tissues. Volume is the main feature used for evaluating the drug/treatment effects, but several other features can be estimated even from a simple 2D image. For high-content screening analysis, the bottleneck is the segmentation stage, which is essential for detecting the spheroids in the images and then proceeding to the feature extraction stage for performing morphotypic analysis. *Problem*: Today, several tools are available for extracting morphological features from spheroid images, but all of them have pros and cons and there is not a general validated solution. Thanks to new deep learning models, it is possible to standardize the process and adapt the analysis to big data. *Novelty*: Starting from the first version of *AnaSP*, an open-source software suitable for estimating several morphological features of 3D spheroids, we implemented a new module for automatically segmenting 2D brightfield images of spheroids by exploiting convolutional neural networks. *Results*: Several deep learning segmentation models (i.e. VVG16, VGG19, ResNet18, ResNet50) have been trained and compared. All of them obtained very interesting results and ResNet18 ranked as the best-performing. *Conclusions*: A network based on an 18-layer deep residual architecture (*ResNet-18*) has been integrated into *AnaSP*, releasing *AnaSP 2.0,* a version of the tool optimized for high-content screening analysis. The source code, standalone versions, user manual, sample images, video tutorial, and further documentation are freely available at: *https://sourceforge.net/p/anasp*.

Keywords: High-content screening; Widefield microscopy; Cancer 3D models; Deep Learning; Morphological Analysis.

1. Introduction

In vivo life is never flat and today we are fully immersed in the three-dimensional (3D) era of modeling [1]. Biology and medicine, like other research fields, are following this trend [2]. 3D multicellular -oids (*e.g.* organoids, spheroids [3]) have become essential in vitro tools for instance for anti-cancer drug screening and toxicology studies in preclinical oncology [4]. Today, multicellular spheroids (*i.e.* multicellular aggregates producing their extracellular matrix and having a nearly spherical shape [5]) can be created from different cell lines with different technologies according to different pros and cons [6]. Once produced, they are then widely used in several high-content screening (HCS) analyses [7], in particular image-based high-throughput experiments [8]. Volume is

the main feature used for instance for evaluating the anti-cancer drug/treatment effects [9], but several other features can be estimated even from a simple 2D image acquired with standard widefield microscopes [10]. For instance, *AMIDA* [11], *AnaSP* [12], *INSIDIA* [13], *PCaAnalyser* [14], *SpheroidJ* [15], *SpheroidSizer* [16], and *TASI* [17] are tools freely available for extracting morphological features of spheroids. The segmentation stage is the bottleneck in this process, as it is required for detecting spheroids in images and extracting features in order to perform wide morphotypic analysis in a limited amount of time [18].

An answer to the segmentation problem can be provided by artificial neural networks (ANNs) specifically trained for segmenting brightfield images of spheroids. An ANN is defined as a computational processing system that is significantly inspired by how biological nervous systems (such as the human brain) work [19]. ANNs are primarily composed of a large number of interconnected computational nodes (referred to as "neurons"), which collaborate in a distributed method to collectively learn from input and optimize their ultimate output [20]. The input, typically a multidimensional vector, is loaded into the input layer, and it is deeply distributed to several hidden layers (today's common term "deep" refers to the number of layers of these networks, at least two). These hidden layers make decisions based on the prior layers, and then weigh how a stochastic change inside itself degrades or improves the final output. This is known as the learning process [21]. The training is typically executed using supervised deep learning techniques, which means that the learning goes through pre-labeled inputs that serve as targets. One of the most common types of ANNs is the convolutional neural network (CNN) [22]. CNNs are similar to regular ANNs: they are made up of "neurons" that optimize themselves through learning. Each neuron continues to accept input and conduct an operation (such as a scalar product followed by a nonlinear function), which is the same foundation of many ANNs. The entire network still expresses a single perceptual scoring function from the input vectors to the final output of the class score (the weight). The final layer includes loss functions related to the classes, as well as all of the standard tips and tricks created for classic ANNs that still apply. The only notable difference between CNNs and traditional ANNs is that CNNs are primarily used in the field of pattern recognition within images. This allows the encoding of image-specific features into the architecture, making the network more suited for image-focused tasks while further reducing the parameters required to set up the model [23].

Today, several deep-learning CNN architectures are available for automatically detecting the objects of interest in the background and segmenting them. Thanks to these deep learning models, it is possible to optimize the segmentation process for adapting the analysis to big data [24]. Between them, the Visual Geometry Group (VGG) network [25] and Residual Network (ResNet) [26] are really promising. VGG is a standard deep CNN architecture with multiple layers. For instance, the popular VGG16 and VGG19 consist of 16 and 19 convolutional layers, respectively [27]. ResNet is a deep learning model designed to solve the so-called "vanishing/exploding gradients" problem, which hampers convergence from the beginning [28]. Similarly to VGG16 and VGG19, the popular ResNet18 and ResNet50 consist of 18 and 50 convolutional layers, respectively.

In this work, firstly, four different deep CNNs were implemented using the *MATLAB Deep Learning Toolbox* (©, The MathWorks, Inc., Natick, MA, USA): VGG16, VGG19, ResNet18, and ResNet50. The networks were trained and compared using the thousands of brightfield images of spheroids, acquired by the different worldwide laboratories contributing to the *MISpheroID* project [29]. Then, a module for easily training new networks was integrated into the open-source tool *AnaSP*, releasing the second version of it. In particular, *AnaSP* 2.0 has been developed in *MATLAB*. It works with *Windows*, *Macintosh*, and *UNIX*-based systems, and it is freely provided with an 18-layer deep residual network (*i.e.*, ResNet18) already integrated. Source code, compiled standalone versions for *Windows*, *Mac,* and *Linux*, a user manual, sample images, a video tutorial, and further documentation are freely available at: *https://sourceforge.net/p/anasp*.

The next sections of this work are organized as follows. **Section 2** presents a short overview of the tools freely available for performing morphotypic analysis of spheroids. **Section 3** describes *AnaSP 2.0* in detail. Finally, **Section 4** reports the main conclusions of the work.

2. Tools for morphotypic analysis

Today, there are several freely available tools for segmenting microscopy images of spheroids and performing morphotypic analysis. This Section reports a brief description of the main features of *AMIDA* [11], *AnaSP* [12], *INSIDIA* [13], *PCaAnalyser* [14], *SpheroidJ* [15], *SpheroidSizer* [16], and *TASI* [17], then summarized in **Table 1** and **Table 2**. **Figure 1** and **Figure 2** report printscreens of the tool's graphical user interface (GUI).

AMIDA [11]: *AMIDA* (*Automated Morphometric Image Data Analysis*, also internally called *VTT Automated Cancer Cell Analysis - ACCA*) is a standalone, freely available software solution for *Windows* systems. It supports large-scale, high-content screens based on 3D cultures. *AMIDA* works with both single and projected images as well as original stacks of confocal images and phase-contrast (PC) images, but in the case of 3D inputs (*x, y, z*), the developers simply decided to restrict the quantitative analysis to 2D maximum projections. *AMIDA* represents a user-friendly tool with a segmentation stage based on two main parameters called sensitivity and threshold. The morphometric features implemented in *AMIDA* can be divided into three classes: (*a*) general, (*b*) morphological, and (*c*) functional. General features include information related to the size (area) of an object, its relation to neighbors (number of neighbors, shared boundaries with neighbors, closest neighbors), and the amount of cellular matter in relation to the local background (cell ratio, average ratio). Morphological features include measures for features typically associated with the phenotype (habitus) of multicellular spheroids, such as symmetry (roundness), contour roughness (measuring small surface features), and measures that indicate invasive processes (appendages). Finally, functional features estimate signal density, the number of cells per structure (cell number), the polarization of cells within the spheroid (''hollowness''), the average size of cells, and the ratio of cells relative to the size of the entire spheroid.

AnaSP [12]: *AnaSP* (*Analysis of SPheroids*) is an open-source tool specifically designed for extracting morphological features starting from the analysis of 2D microscopy images of 3D multicellular spheroids. It works with *Windows*, *Macintosh*, and *UNIX*-based systems. In particular, it directly works with gray-level PC, differential interference contrast (DIC), and brightfield images generally characterized by a light background, but it can easily work even with fluorescent images just pre-inverted (to have a final light background) and converted to gray-level using common, freely available external tools like *ImageJ* [30] and *Fiji* [31]. *AnaSP* was created with the idea of providing an extremely user-friendly tool where the user can manually, semi-automatically, and automatically segment the spheroids, always having the opportunity to easily correct the segmentation obtained. *AnaSP* is a perfect solution for those who want to perform HCS experiments. The masks obtained (*i.e.*, black and white masks with value *1* assigned to the pixels belonging to the foreground pixels of the spheroid) are automatically saved as output. They are then used for extracting the morphological features such as minor and major axes, equivalent diameter, perimeter, area, sphericity, and even the 3D volume [32], which are computed simply by analyzing 2D images [33], according to the *ReViSP* algorithm [34]. In case of more spheroids present in the image, *AnaSP* by default analyses the largest one, although a working modality considering more spheroids of similar size can be selected. The features computed are simply exported as *Microsoft Excel* tables, ready for further statistical analysis.

INSIDIA [13]: *INSIDIA* (*INvasion SpheroID ImageJ Analysis*) is an open-source code implemented as a customizable macro running on *Fiji*. It enables HCS quantitative analysis of gray-level spheroid images with the output of a range of parameters defining the spheroid core and its invasive

characteristics. *INSIDIA* works with both 2D fluorescent and brightfield images. However, programming experience is required to be able to modify the original macro written in *Java*. The macro isolates the entire spheroid cellular mass from the image background with several user options able to address poorly contrasted images. *INSIDIA* distinguishes the spheroid core from the invasive edge and provides quantitative information describing growth and invasive behavior, for instance: the center of the spheroid, diameters, total area, envelope area, perimeter, circularity, shape factor, surface irregularity, and roughness. *INSIDIA* analysis is based on three sequential steps: (*a*) spheroid segmentation; (*b*) density profile / core-thresholding analysis; (*c*) density map analysis. The segmentation comprises a set of morphological operations that isolate the spheroid from the image background. Briefly, it is possible to declare that *INSIDIA* has two main segmentation options (a threshold-based and a Frangi-filtering-based one [35]), but the user does not have the opportunity to manually correct specific parts of the segmentation but just to adjust parameters that influence the entire contour. The output is a binary mask with a foreground intensity of *255* (white pixels) and a background intensity of *0* (black pixels). The mask is then internally used to distinguish and quantify the spheroid's core from the invasive edge.

PCaAnalyser [14]: *PCaAnalyser* (*Prostate Cancer Analyser*) is a *Java*-based program developed as an *ImageJ* plugin. It operates on 2D fluorescent images in one of several *Bio-Formats* [36]. The analysis undertaken by *PCaAnalyser* is composed of two major algorithmic interfaces. In the first step, the boundary of the cellular 3D spheroid is detected and the required masks are generated. In the second step, nuclei are detected and spheroid memberships are then predicted using the masks and the boundaries. Similar approaches are followed to detect and study cytoplasmic areas by segregating them from critical noise. The paradigm of *PCaAnalyser*, including the reporting component, has been designed to be flexible to enable the user to readily manipulate related analysis in a variety of ways, in addition to the default options. Concerning the efficiency of *PCaAnalyser*, a candidate-membership-based algorithm has been incorporated to speed up the nucleus-spheroid detection process, making the overall processing time considerably faster. As currently configured, the *PCaAnalyser* can quantify a range of morpho-biological parameters including the number of spheroids per image, the number of nuclei per spheroid, perimeter of the nucleus, nuclei-spheroid membership prediction, various function-based classifications of peripheral and non-peripheral areas to measure the expression of biomarkers and protein constituents, as well as effectively defining segregated cellular-objects for a range of signal-to-noise ratios. In addition, the *PCaAnalyser* architecture is highly flexible, operating as a single independent analysis, as well as in batch mode, which is essential for HCS.

SpheroidJ [15]: *SpheroidJ* is a user-friendly, open-source *ImageJ/Fiji* plugin for segmenting PC, DIC, brightfield, and fluorescent images of spheroids. It is written in *Java* and *Python*. It is platform-independent and requires the installation of a few external libraries, but only for using the provided deep learning model. Given an image containing a spheroid, *SpheroidJ* aims to produce a mask for the region that contains it. The authors implemented several segmentation strategies for different scenarios. They are based on the sequential application of several image processing techniques, such as edge detection or thresholding, and morphological operations like dilation or erosion. Specifically, the procedure can be split into two steps, contour generation, and contour refinement, and the user must define just five different parameters: (*a*) the number of iterations for the edge detector; (*b*) the thresholding method; (*c*) the number of iterations for the dilation and erosion operations; (*d*) a flag for enabling the fill holes operation; (*e*) a flag for enabling the watershed operation. However, those algorithms fail to generalize different conditions. By contrast, the deep learning model provided, constructed using the *HRNet-Seg* architecture, generalizes properly to a diversity of scenarios, but unfortunately, there is no procedure to train new deep learning models. In addition, *SpheroidJ* does not provide a simple way of visualizing and editing the segmentation results, and the features computed are limited to just area, area fraction, perimeter, circularity, Feret diameters, and Feret angles.

SpheroidSizer [16]: *SpheroidSizer* is an open-source, high-throughput image analysis script with GUI. It is optimized to process different batches of images of spheroids in the same session, but it currently works with just 2D brightfield images, each containing a single spheroid. Today, there is no ready-to-use standalone version of the tool, and the script for running it requires a *MATLAB* license to work, including licenses for the *Signal Processing Toolbox*, the *Image Processing Toolbox*, and the *Parallel Computing Toolbox*. The segmentation method used is adapted from the well-known Snakes active contour algorithm [37], which is especially suitable for images with uneven illumination and noisy background. It can also tolerate many usual artifacts, such as debris, originating from the specimen. However, the complimentary "*Manual Initialize*" and "*Hand Draw*" tools provide flexibility to *SpheroidSizer* in dealing with various types of spheroids and diverse quality images. *SpheroidSizer* just measures the major axis, minor axis, and volume (computed simply as *V = 0.5 * major axis * minor axis * minor axis*) of each spheroid, and provides the results in an output spreadsheet for easy manipulations in the subsequent data analysis.

TASI [17]: *TASI* (*Temporal Analysis of Spheroid Imaging*) is a *MATLAB* script that allows investigators to objectively characterize spheroid growth and invasion dynamics. *TASI* is provided as open-source code without any GUI and as a standalone version. Accordingly, it requires a *MATLAB* license and programming experience to work. *TASI* performs spatiotemporal segmentation of spheroid cultures, extraction of features describing spheroid morpho-phenotypes, mathematical modeling of spheroid dynamics, and statistical comparisons of experimental conditions. It was designed for 3D (*x, y, z*) fluorescent images acquired in time-lapse, practically fluorescent 4D datasets (*x, y, z, t*). However, in the first internal step, the input 4D data is immediately reduced to a time sequence of 2D images (*x, y, t*). In particular, *TASI* currently uses a projection operation to reduce the *z* dimension to simplify image analysis. Then, the time sequence of 2D images is processed using an energy-minimizing graph cut segmentation algorithm to automatically delineate spheroid boundaries [38], but no manual opportunity is available for correcting parts of the segmentation. The masks obtained are then used for extracting the spatiotemporal features to describe spheroid growth, shape, and motion. In particular, *TASI* extracts several basic morphology features including area, perimeter, eccentricity, and intensity statistics. The complexity of the spheroid boundary is also measured. To quantify the spheroid's branching behavior, a "core radius" that captures the size of the main spheroid mass and an "invasive radius" that captures the extent of the projections are then computed. In particular, the core radius was defined as the radius of the largest circle that can be inscribed within the spheroid mask, centered at the mask centroid. The invasive radius was defined as the minimum circle that can encompass the entire spheroid, including any invasive branches. These radii roughly capture growth due to proliferation and growth due to invasion. The number of branches was further quantified using a skeletonization procedure. Morphological operations were applied to thin the mask to a skeletal structure, and the terminal endpoints were counted. This process robustly captures the tips of branching structures, even with complex shapes.

Fig. 1. Printscreen of the GUI of *AnaSP 2.0*. The tool has been extended with several functionalities, including different deep-learning segmentation models and new morphological features.
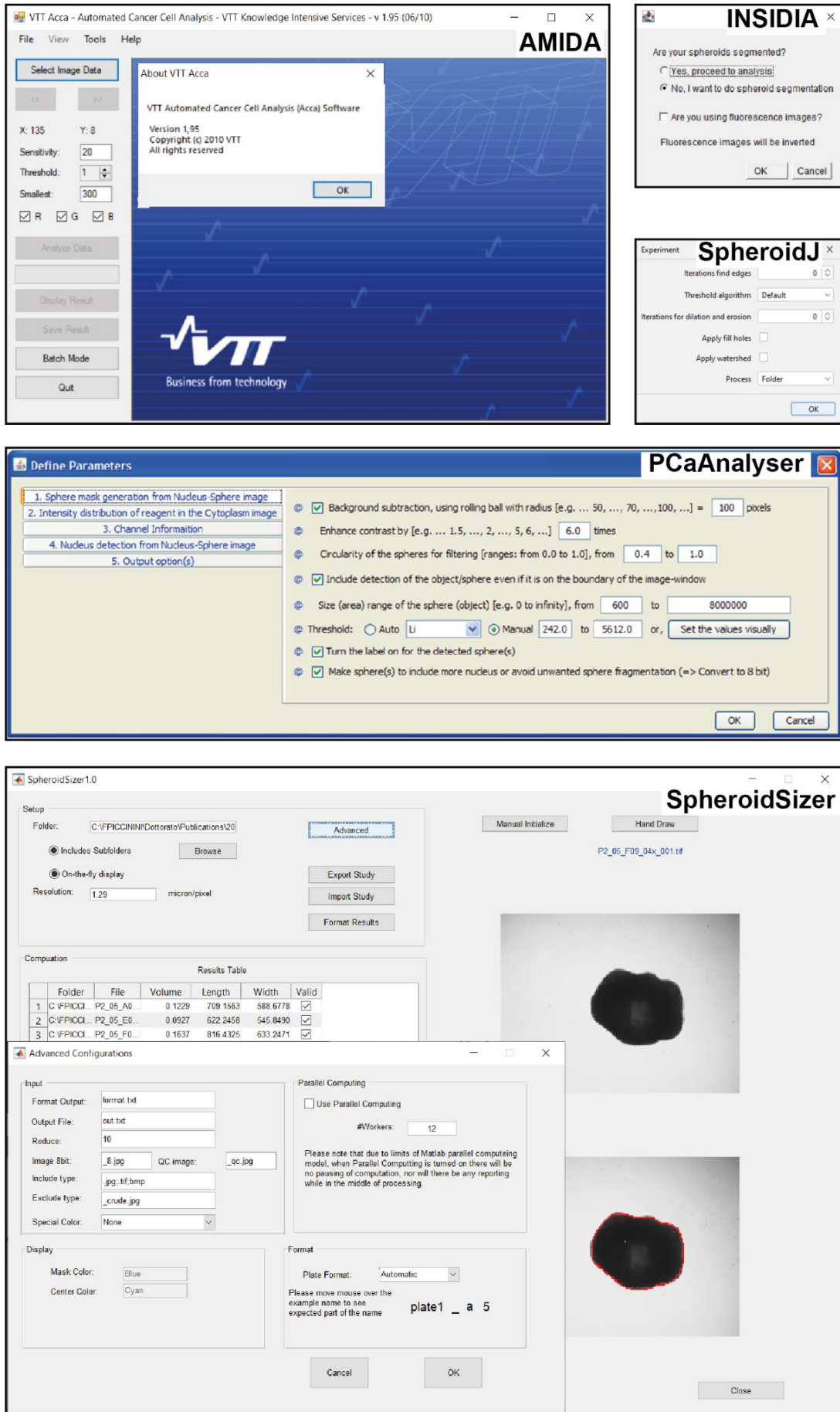
Fig. 2. Printscreens of the different tools having a GUI.

Table 1: Freely available tools for performing morphotypic analysis of spheroids - characteristics (X = available/yes; O = not available/no).

	AMIDA	AnaSP	INSIDIA	PCaAnalyser	SpheroidJ	SpheroidSizer	TASI
VERSION							
Year first release	2010	2015	2017	2013	2020	2013	2015
Current version	1.95	2.0	2022	2013	1.0.1	2013	2018
DOCUMENTATION							
User guide	O	X	X	X	O	X	X
Website	O	X	X	O	X	O	X
Video tutorial	O	X	O	O	O	X	O
Sample dataset	X	X	X	O	X	O	O
Open source	O	X	X	X	X	X	O
Implementation language	C#	MATLAB	Java	Java	Java/Python	MATLAB	MATLAB
USABILITY							
Input image format	tif	All common	All common	All common	All common	tif, jpg, bmp	tif
No programming experience required	X	X	O	X	X	O	O
User-friendly GUI	X	X	X	X	X	X	O
Intuitive visualisation settings	X	X	X	X	X	X	X
No commercial licences required	X	X	X	X	X	O	O
Portability on Win/Linux/Mac	Only Win	X	X	X	X	Only Win	Only Win
FUNCTIONALITY							
Image pre-processing	X	O	O	X	O	O	O
Brightfield/DIC/PC images	X	X	X	O	X	X	X
Fluorescence images	X	O	X	X	O	O	O
Manual segmentation	O	X	O	X	O	X	O
Automatic segmentation	X	X	X	X	X	X	X
Feature definition	O	X	O	O	O	O	O
OUTPUT							

Segmentations	X	X	X	X	X	X	X
Features	X	X	X	X	X	X	X
Statistics	X	O	O	X	O	O	X

Table 2: Freely available tools for performing morphotypic analysis of spheroids - download link (last access 31/12/2022).

AMIDA	https://doi.org/10.1371/journal.pone.0096426.s010
AnaSP	https://sourceforge.net/p/anasp
INSIDIA	https://valentinapalmieri.wixsite.com/insidia
PCaAnalyser	https://www.dropbox.com/s/qefb7bgo9wvranr/SuppMaterial.zip
SpheroidJ	https://github.com/joheras/SpheroidJ
SpheroidSizer	https://www.jove.com/files/ftp_upload/51639/SpheroidSizerJOVE.zip
TASI	http://github.com/cooperlab/TASI

3. AnaSP 2.0

3.1. New functionalities

Starting from the first version of *AnaSP*, we developed *AnaSP 2.0* by integrating deep learning modules for training and using pre-trained networks to segment spheroids in PC, DIC, and brightfield images. In particular, (*a*) four different deep learning architectures are now available for training new segmentation models: VGG16, VGG19, ResNet18, and ResNet50. In addition, all the old functionalities in *AnaSP 1.0* have been maintained, and most of them have been extended: (*b*) The input images can now be gray-level and RGB tif, tiff, png, bmp, and jpg images, with 8, 12 and 16 bits; (*c*) The manual editing opportunity has been extended by introducing new opportunities connected to the mouse's buttons (*e.g.* drag&drop, color change and deletion of the drawn contour); (*d*) The number of features estimated is now fifteen and precisely they are: (*1*) area, (*2*) circularity, (*3*) compactness, (*4*) convexity, (*5*) equivalent diameter, (*6*) Feret aspect ratio, (*7*) Feret diameter max, (*8*) Feret diameter max orthogonal distance, (*9*) Feret diameter min, (*10*) length major diameter through centroid, (*11*) length minor diameter through centroid, (*12*) perimeter, (*13*) solidity, (*14*) sphericity, (*15*) volume. In addition, the procedure for easily defining new (or customized) features just by using a model template has been maintained. Definitions/equations of the currently estimated features and a detailed description of the procedure to define new features are reported in the User Manual. **Figure 1** shows a printscreen of *AnaSP 2.0*'s GUI.

3.2. Validation

3.2.1 Training

To validate the deep-learning modules, we trained the 4 different networks, VGG16, VGG19, ResNet18, and ResNet50. A set of approximately 10000 2048x2048 RGB spheroid images and relative masks was used. The dataset was divided into 3 subsets: (*a*) the training set (80% of the total

images); (*b*) the validation set (10% of the total images); (*c*) the test set (10% of the total images). To reduce computational efforts, all the input images were rescaled to 8-bit gray-level 500x500 pixels and the masks were internally binary labeled into 2 categories, background and foreground (*i.e.*, the spheroid). All the networks have been trained considering 7 epochs, with 20 batches per epoch and an initial learning rate of 0.001 that changes according to a piecewise learning rate schedule with a period of 2 and a factor of 0.5. The training dataset is shuffled after every epoch. Due to the different characteristics of the networks, the training time was different: VGG16 required approximately 6 hours, VGG19 10 hours, ResNet18 15 hours, and ResNet50 20 hours. However, it is worth noting that the computational time and complexity related to the usage of the different trained networks are comparable.

3.2.2 Validation

The masks obtained with the 4 trained networks were compared with the segmentations obtained using *AnaSP* 1.0 (**Figure 3**).
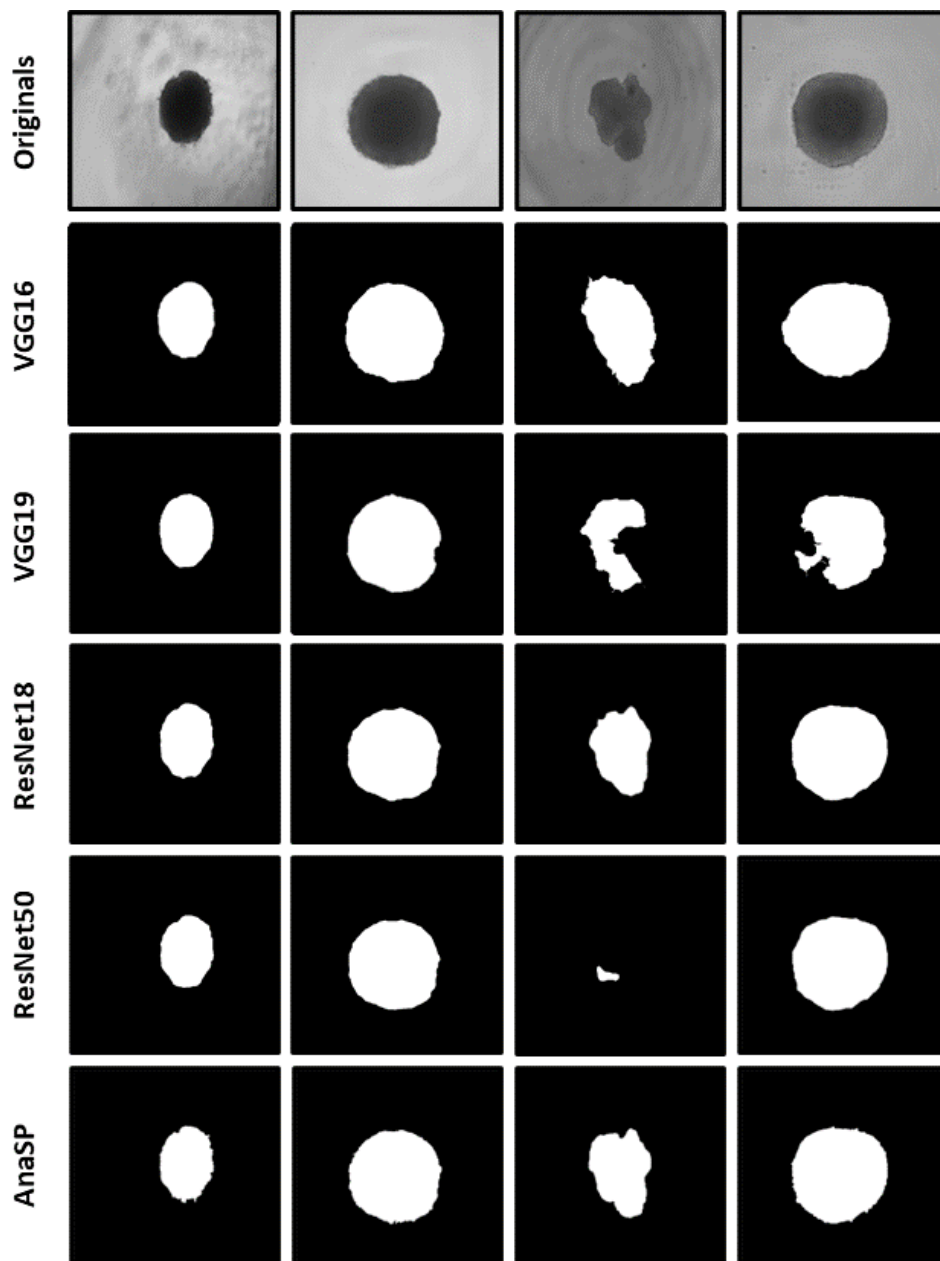
Fig. 3: Top line: original images. Middle lines: masks obtained with the different trained networks. Bottom line: masks obtained with the threshold-based algorithm traditionally available in AnaSP.

Normalized confusion matrix, *mean accuracy* (MA) and *average intersection over union* (IoU, [39]) were computed. The *Normalized confusion matrix* is a matrix that shows on its diagonal the percentage of pixels correctly classified as background and foreground, and outside of the diagonal, the percentage of misclassified pixels. *Mean accuracy* is the ratio of correctly classified pixels in each class to the total number of pixels, averaged over all the classes. *Average IoU* is the area of overlap between the predicted segmentation and the reference one, divided by the area of union between the predicted segmentation and the reference one. This parameter is calculated by averaging the IoU values of the different classes. In numbers, IoU is defined as the ratio between TP (true positives) and the sum of TP (true positives), FP (false positives), and FN (false negatives), where TP are the foreground pixels classified in the correct class; FP are the pixels classified as foreground but background in the reference image; FN are the pixels classified as background but foreground in the reference image.

3.2.3 Comparison with AnaSP 1.0

According to the different metrics, all the trained networks obtained nice and pretty comparable scores (**Table 3**).

Table 3: Metric values obtained during the network validation.

	VGG16	VGG19	ResNet18	ResNet50
TP	0.99	0.99	0.99	0.99
TN	0.91	0.81	0.91	0.74
FP	0.09	0.19	0.09	0.26
FN	0.01	0.01	0.01	0.01
MA	0.97	0.93	0.95	0.87
IoU	0.91	0.90	0.92	0.86

Firstly, by looking at the TP values, it is evident that all the trained networks perfectly segmented the foreground pixels. Considering the FP values, it is possible to see that VGG19 and ResNet50 misclassified 20-30% of background pixels. On the other hand, VGG16 and ResNet18 show higher performances also for background segmentation, having comparable results with each other. Looking at the *MA* and the *average IoU*, it is clear that ResNet50 was the worst, while the others appeared quite similar. Combining the information, it resulted that VGG16 and ResNet18 were the most reliable networks. Obtaining better results for ResNet18 with respect to ResNet50 is something not surprising: the dataset is composed just of images with "black" spheroids on a "white" background. Especially for these cases, increasing complexity and depth (i.e. number of layers) of artificial neural networks do not necessarily mean having better segmentations due to overfitting and many other problems [40]. However, some general qualitative considerations can be made and **Figure 4** shows some representative images. For example, it is worth remarking that the segmentation often fails with spheroids that have a necrotic core. This is due to the fact that the center of the spheroid in the image is the same color as the background, making it almost impossible for the network to recognize it. In general, the trained networks showed difficulties segmenting images in which the spheroids have a color that is similar to the background. When the spheroid touches the image border, the

segmentation typically fails. Then, VGG16 typically fails if the background is not uniform. Concluding, ResNet18 is the one with the best overall performance in segmenting spheroids. Accordingly, we released *AnaSP* 2.0 by including the trained ResNet18.
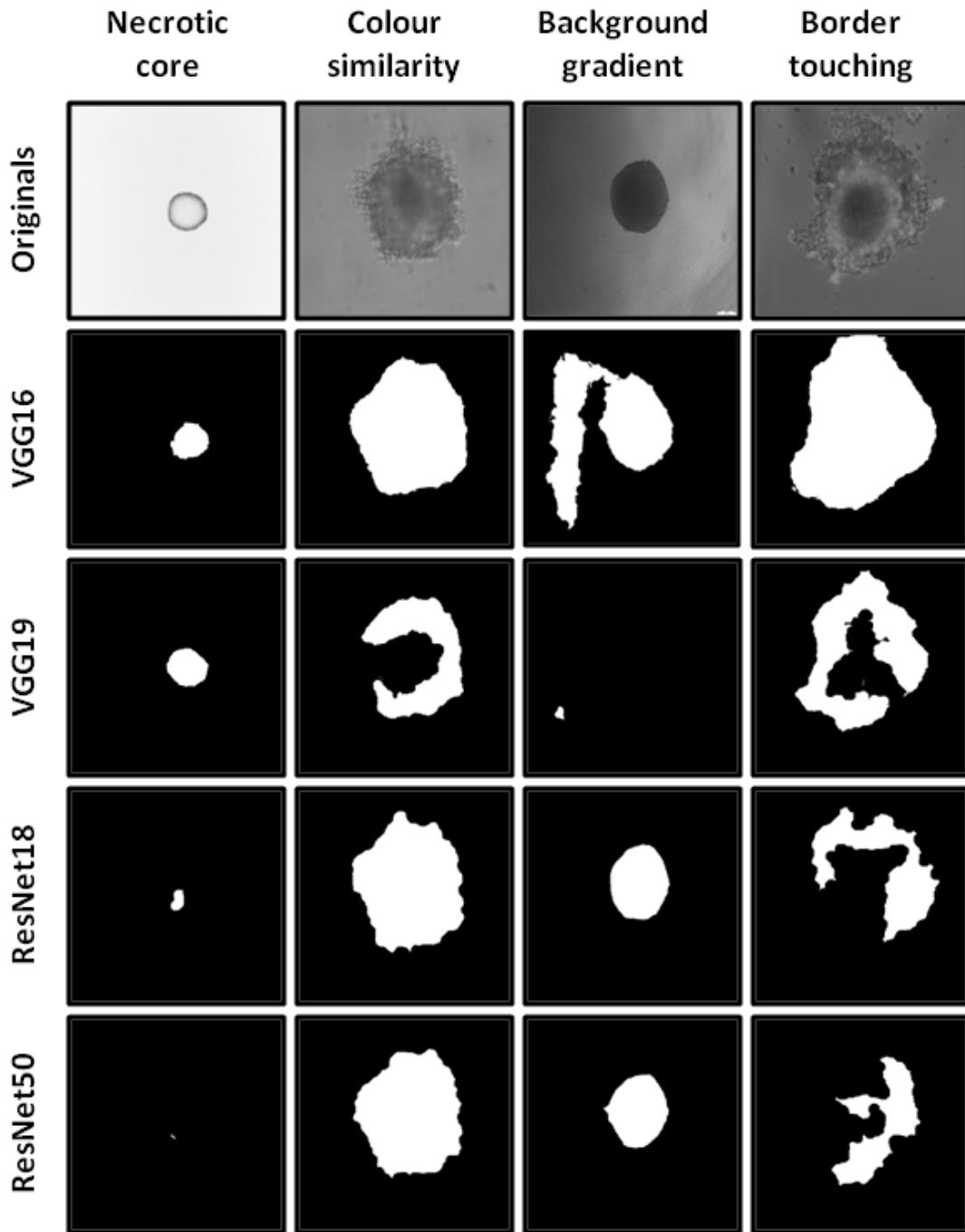

Fig. 4: Top line: original images representative of different segmentation problems. Other lines: masks obtained with the different trained networks.

4. Conclusions

Today, cancer multicellular spheroids are widely used in several HCS experiments of chemotherapeutic drugs and radiotherapy treatments. Several tools are available for extracting morphological features and performing morphotypic analysis. The bottleneck of the process is the

segmentation stage, but CNNs specifically trained for segmenting images of spheroids can solve the problem.

In this work, four different deep CNNs (*i.e.* VVG16, VGG19, ResNet18, ResNet50) were implemented, trained and tested. All of them obtained very interesting results and ResNet18 ranked as the best-performing. Accordingly, besides designing and integrating into *AnaSP* a new module for easily training new networks, we released a new version of it (*i.e. AnaSP 2.0*) by directly including a ResNet18 trained by using the thousands of images acquired by the different laboratories contributing to the *MISpheroID* project, and available for accurately segmenting new brightfield images of spheroids.

AnaSP 2.0 is distributed as an open-source tool, freely available at: *https://sourceforge.net/p/anasp*.

Author Contributions
Conceptualization: F.P., A.P., M.S., J.C.P., O.D.W., A.T., G.M., G.C.; Data curation: F.P., A.P., M.S., J.C.P., M.M.T., M.T.; Formal analysis: F.P., A.P., M.S., J.C.P., M.T., O.D.W., A.T.; Investigation: F.P., A.P., M.S., M.M.T.; Methodology: F.P., A.P., M.S., J.C.P.; Project administration: F.P., J.C.P., M.T., O.D.W., A.T., G.M., G.C.; Validation: F.P., A.P., M.S., J.C.P.; Visualization: F.P., A.P., M.S., J.C.P.; Writing - original draft: F.P., A.P., M.S., J.C.P., M.M.T.; Writing - review & editing: M.T., O.D.W., A.T., G.M., G.C.

Conflicts of Interest
There are no conflicts to declare.

References
[1] Fitzgerald KA, Malhotra M, Curtin CM, O'Brien FJ, O'Driscoll CM, Life in 3D is never flat: 3D models to optimise drug delivery. *Journal of Controlled Release* 215:39-54, 2015.
[2] Sakalem ME, De Sibio MT, da Costa FADS, de Oliveira M. Historical evolution of spheroids and organoids, and possibilities of use in life sciences and medicine. *Biotechnology Journal* 16(5):2000463, 2021.
[3] Tasnadi EA, Toth T, Kovacs M, Diosdi A, Pampaloni F, Molnar J, Piccinini F, Horvath P. 3D-Cell-Annotator: an open-source active surface tool for single-cell segmentation in 3D microscopy images. *Bioinformatics* 36(9):2948-2949, 2020.
[4] Gunti S, Hoke AT, Vu KP, London Jr NR. Organoid and spheroid tumor models: Techniques and applications. *Cancers* 13(4):874, 2021.
[5] Piccinini F, De Santis I, Bevilacqua A. Advances in cancer modeling: fluidic systems for increasing representativeness of large 3D multicellular spheroids. *Biotechniques* 65(6):312-314, 2018.
[6] Costa EC, Moreira AF, de Melo-Diogo D, Gaspar VM, Carvalho MP, Correia IJ. 3D tumor spheroids: an overview on the tools and techniques used for their analysis. *Biotechnology Advances* 34(8):1427-1441, 2016.
[7] Carragher N, Piccinini F, Tesei A, Trask Jr OJ, Bickle M, Horvath P. Concerns, challenges and promises of high-content analysis of 3D cellular models. *Nature Reviews Drug Discovery* 17(8):606, 2018.
[8] Lin S. Schorpp K, Rothenaigner I, Hadian K. Image-based high-content screening in drug discovery. *Drug Discovery Today* 25(8):1348-1361, 2020.
[9] De Santis I, Tasnadi E, Horvath P, Bevilacqua A, Piccinini F. Open-source tools for volume estimation of 3D multicellular aggregates. *Applied Sciences* 9(8):1616, 2019.

[10] Celli JP, Rizvi I, Blanden AR, Massodi I, Glidden MD, Pogue BW, Hasan T. An imaging-based platform for high-content, quantitative evaluation of therapeutic response in 3D tumour models. *Scientific Reports* 4(1):1-10, 2014.

[11] Härmä V, Schukov HP, Happonen A, Ahonen I, Virtanen J, Siitari H, Akerfelt M, Lotjonen J, Nees M. Quantification of dynamic morphological drug responses in 3D organotypic cell cultures by automated image analysis. *PloS One* 9(5):e96426, 2014.

[12] Piccinini F. AnaSP: a software suite for automatic image analysis of multicellular spheroids. *Computer Methods and Programs in Biomedicine* 119(1):43-52, 2015.

[13] Moriconi C, Palmieri V, Di Santo R, Tornillo G, Papi M, Pilkington G, De Spirito M, Gumbleton M. INSIDIA: A FIJI Macro Delivering High-Throughput and High-Content Spheroid Invasion Analysis. Biotechnology Journal 12(10):1700140, 2017.

[14] Hoque MT, Windus LC, Lovitt CJ, Avery VM. PCaAnalyser: A 2D-image analysis based module for effective determination of prostate cancer progression in 3D culture. *PLoS One* 8(11):e79865, 2013.

[15] Lacalle D, Castro-Abril HA, Randelovic T, Domínguez C, Heras J, Mata E, Mata G, Méndez Y, Pascual V, Ochoa I. SpheroidJ: an open-source set of tools for spheroid segmentation. *Computer Methods and Programs in Biomedicine* 200:105837, 2021.

[16] Chen W, Wong C, Vosburgh E, Levine AJ, Foran DJ, Xu EY. High-throughput image analysis of tumor spheroids: a user-friendly software application to measure the size of spheroids automatically and accurately. *JoVE (Journal of Visualized Experiments)* 89:e51639, 2014.

[17] Hou Y, Konen J, Brat DJ, Marcus AI, Cooper LA. TASI: A software tool for spatial-temporal quantification of tumor spheroid dynamics. *Scientific Reports* 8(1):1-9, 2018.

[18] Chen W, Wong C, Vosburgh E, Levine AJ, Foran DJ, Xu EY. High-throughput image analysis of tumor spheroids: a user-friendly software application to measure the size of spheroids automatically and accurately. *JoVE (Journal of Visualized Experiments)* 89:e51639, 2014.

[19] Dongare AD, Kharde RR, Kachare AD. Introduction to artificial neural network. *International Journal of Engineering and Innovative Technology (IJEIT)* 2(1):189-194, 2012.

[20] Maind SB, Wankar P. Research paper on basic of artificial neural network. *International Journal on Recent and Innovation Trends in Computing and Communication* 2(1):96-100, 2014.

[21] Zou J, Han Y, So SS. Overview of artificial neural networks. *Artificial Neural Networks* 14-22, 2008.

[22] Albawi S, Mohammed TA, Al-Zawi S. Understanding of a convolutional neural network. *In Proceedings of the IEEE International Conference on Engineering and Technology (ICET)* 1-6, 2017

[23] O'Shea K, Nash R. An introduction to convolutional neural networks. *arXiv preprint* 1511.08458, 2015.

[24] Lacalle D, Castro-Abril HA, Randelovic T, Domínguez C, Heras J, Mata E, *et al.*, Ochoa I. SpheroidJ: an open-source set of tools for spheroid segmentation. *Computer Methods and Programs in Biomedicine* 200:105837, 2021.

[25] Hammad I, El-Sankary K. Impact of approximate multipliers on VGG deep learning network. *IEEE Access* 6:60438-60444, 2018.

[26] He K, Zhang X, Ren S, Sun J. Deep residual learning for image recognition. *In Proceedings of the IEEE conference on Computer Vision and Pattern Recognition (CVPR)* 770-778, 2016.

[27] Vedaldi A, Zisserman A. Vgg convolutional neural networks practical. *Department of Engineering Science, University of Oxford* 66, 2016.

[28] He K, Zhang X, Ren S, Sun, J. Identity mappings in deep residual networks. *In Proceedings of the European Conference on Computer Vision (ECCV)* 630-645, 2016.

[29] Peirsman A, Blondeel E, Ahmed T, Anckaert J, Audenaert D, Boterberg T, *et al.*, De Wever O, MISpheroID: a knowledgebase and transparency tool for minimum information in spheroid identity. *Nature Methods* 18:1294-1303, 2021.

[30] Collins T J. ImageJ for microscopy. *Biotechniques* 43(S1):S25-S30, 2007.

[31] Schindelin J, Arganda-Carreras I, Frise E, Kaynig V, Longair M, Pietzsch T, Preibisch S, Rueden C, Saalfeld S, Schmid B, Tinevez JY, White DJ, Hartenstein V, Eliceiri K, Tomancak P, Cardona A. Fiji: an open-source platform for biological-image analysis. *Nature Methods* 9(7):676-682, 2012.

[32] De Santis I, Tasnadi E, Horvath P, Bevilacqua A, Piccinini F. Open-source tools for volume estimation of 3D multicellular aggregates. *Applied Sciences* 9(8):1616, 2019.

[33] Piccinini F, Tesei A, Bevilacqua A, Single-image based methods used for non-invasive volume estimation of cancer spheroids: a practical assessing approach based on entry-level equipment. *Computer Methods and Programs in Biomedicine* 135:51-60, 2016.

[34] Piccinini F, Tesei A, Arienti C, Bevilacqua A, Cancer multicellular spheroids: Volume assessment from a single 2D projection. *Computer Methods and Programs in Biomedicine* 118(2):95-106, 2015.

[35] Frangi AF, Niessen WJ, Hoogeveen RM, Van Walsum T, Viergever MA. Model-based quantitation of 3-D magnetic resonance angiographic images. *IEEE Transactions on Medical Imaging* 18(10):946-956, 1999.

[36] Schneider CA, Rasband WS, Eliceiri KW. NIH Image to ImageJ: 25 years of image analysis. *Nature Methods* 9(7):671-675, 2012.

[37] Chan TF, Vese LA. Active contours without edges. *IEEE Transactions on Image Processing* 10(2):266-277, 2001.

[38] Yuan J, Bae E, Tai XC. A study on continuous max-flow and min-cut approaches. *In Proceedings of the IEEE conference on Computer Vision and Pattern Recognition (CVPR)* 2217-2224, 2010.

[39] Piccinini F, Balassa T, Carbonaro A, Diosdi A, Toth T, Moshkov N, Tasnadi EA, Horvath P. Software tools for 3D nuclei segmentation and quantitative analysis in multicellular aggregates. Computational and Structural Biotechnology Journal, 18, 1287-1300, 2020.

[40] Bressem KK, Adams LC, Erxleben C, Hamm B, Niehues SM, Vahldiek JL. Comparing different deep learning architectures for classification of chest radiographs. Scientific Reports, 10(1), 13590, 2020.