applied sciences

MDPI

Article

Evolutionary Grid Optimization and Deep Learning for Improved In Vitro Cellular Spheroid Localization

Jonas Schurr [1,2,*,†], Hannah Janout [1,2,†], Andreas Haghofer [1,2], Marian Fürsatz [3,4], Josef Scharinger [5], Stephan Winkler [1,2] and Sylvia Nürnberger [3,4]

1 Bioinformatics Research Group, University of Applied Sciences Upper Austria, Softwarepark 11-13, 4232 Hagenberg, Austria
2 Department of Computer Science, Johannes Kepler University, Altenberger Straße 69, 4040 Linz, Austria
3 Ludwig Boltzmann Institute for Traumatology, The Research Center in Cooperation with the AUVA Austrian Cluster for Tissue Regeneration, 1200 Vienna, Austria
4 Department of Orthopedics and Trauma-Surgery, Division of Trauma-Surgery, Medical University of Vienna, 1090 Vienna, Austria
5 Institute of Computational Perception, Johannes Kepler University, Altenberger Straße 69, 4040 Linz, Austria; josef.scharinger@jku.at
* Correspondence: jonas.schurr@fh-hagenberg.at
† These authors contributed equally to this work.

Abstract: The recently developed high-throughput system for cell spheroid generation (SpheroWell) is a promising technology for cost- and time-efficient in vitro analysis of, for example, chondrogenic differentiation. It is a compartmental growth surface where spheroids develop from a cell monolayer by self-assembling and aggregation. In order to automatize the analysis of spheroids, we aimed to develop imaging software and improve the localization of cell compartments and fully formed spheroids. Our workflow provides automated detection and localization of spheroids in different formation stages within Petri dishes based on images created with a low-budget camera imaging setup. This automated detection enables a fast and inexpensive analysis workflow by processing a stack of images within a short period of time, which is essential for the extraction of early readout parameters. Our workflow combines image processing algorithms and deep learning-based image localization/segmentation methods like Mask R-CNN and Unet++. These methods are refined by an evolution strategy for automated grid detection, which is able to improve the overall segmentation and classification quality. Besides the already pre-trained neural networks and predefined image processing parameters, our evolution-based post-processing provides the required adaptability for our workflow to deliver a consistent and reproducible quality. This is especially important due to the use of a low-budget imaging setup with various light conditions. The to-be-detected objects of the three different stages show improved results using our evolutionary post-processing for monolayer and starting aggregation with Dice coefficients of 0.7301 and 0.8562, respectively, compared with the raw scores of 0.2879 and 0.8187. The Dice coefficient of the fully formed spheroids in both cases is 0.8829. With our algorithm, we provide automated analyses of cell spheroid by self-assembling in SpheroWell dishes, even if the images are created using a low-budget camera setup.

Keywords: heuristic optimization; evolution strategy; computer vision; deep learning; machine learning; image processing; spheroids; quantification; high-throughput screening

Citation: Schurr, J.; Janout, H.; Haghofer, A.; Fürsatz, M.; Scharinger, J.; Winkler, S.; Nürnberger, S. Evolutionary Grid Optimization and Deep Learning for Improved In Vitro Cellular Spheroid Localization. *Appl. Sci.* **2024**, *14*, 9476. https://doi.org/10.3390/app14209476

Academic Editor: Pedro Couto

Received: 20 September 2024
Revised: 13 October 2024
Accepted: 15 October 2024
Published: 17 October 2024

1. Introduction

Cell spheroids as a 3D in vitro model are highly important for primary as well as translational research and considered the gold standard, especially for chondrogenic differentiation [1,2]. One major drawback of standard spheroid formation methods is the time and costs required for the procedure steps, as well as the reagents required to generate and cultivate the spheroids and generate the final readouts. Furthermore, the cultivation time

for chondrogenicity is quite long, but there is no possibility of generating early readouts before the endpoint analysis. To address this problem, a new technology (SpheroWell) has been developed that enables the creation of multiple spheroids from cell monolayers by self-assembling within a single vessel by segregating the growth surface of a standard cell culture surface such as Petri dishes [3]. Additionally, media consumption is reduced through the self-assembly of cell monolayers.

The growth surface is segregated into single compartments by a CNC-guided CO_2 laser. The engravings produce a grid-like structure of repeating equidistant crossed lines spanning a grid containing multiple compartments. The engraved surface possesses anti-adhesive properties to prohibit the compartments' attachment, allowing self-assembly. Since the culture surface is flat standard culture plastic and not curved or intransparent as in many spheroid systems, it is possible to monitor the assembly of single spheroids over time.

With (microscopy) imaging, the resulting spheroids and formation behavior can be assessed to identify the effect of treatments like anti-inflammatory compounds. Systematic quantitative assessment of spheroid formation stages can be highly tedious and time-consuming due to large amounts of spheroids over large numbers of images. An automated approach is crucial to extracting knowledge about the formation stage and important metrics efficiently during these experiments.

Macroscopic images of Petri dish plates that were seeded with adipose-derived stem-cells were used to monitor the spheroid formation process over time. The images were taken using a digital of the shelf camera (Olympus OM-D E-M1).

The spheroid formation process can be separated into multiple states, which is significant, as this differentiation facilitates the possibility of detecting treatment effects in early-stage readouts compared with only identifying fully formed spheroids. The three different states of spheroid formation are referred to and described as follows [3]:

1. A monolayer is described as a compartment with a fully attached cell layer filling 100% of the compartment.
2. The stage of formation (aggregation) is described by compartments with cells in the interim state between fully attached (monolayer) and fully formed (spheroid). The aggregation is described as a compartment where >50% of the cell layer remains attached. The formation of the late stage is described as a compartment where <50% of the cell layer remains attached but is not yet fully developed (not circle-shaped).
3. Fully formed is described as a circle-shaped spheroid. Due to detachment, spheroids can also move to other compartments, and two or more spheroids can appear in one compartment.

Our workflow combines state-of-the-art methods based on image processing and artificial neural networks with a reliable and adaptive extraction of the grid structure in SpheroWell Petri dishes through an evolution strategy. Combining these methods provides an improved and automatic methodology for quantifying the spheroid stages in the compartments of the Petri dish images. As described in the methods section, several steps are necessary to enable automatic information extraction and subsequent quantification. These include segmenting and classifying a cell compartment's content to extract one of the three classes mentioned above. Furthermore, evolution-based grid extraction reduces segmentation errors, improving the resulting accuracy of the segmentation and classification results.

2. State of the Art for In Vitro Cell Detection

Improvements of in vitro testing setups and screening systems for cell spheroids using SpheroWell Petri dishes have created new challenges for analysis and extraction. Manually performed steps for an analysis of the cells, such as localization and quantification, are time-consuming and/or prone to errors caused by multiple algorithms carried out in different software, such as those provided by microscope manufacturers or ImageJ [4].

Technological advancements have made it possible to generate image data faster with simpler setups and a higher experiment frequency. However, analyzing a large number of images increased the need for better analysis workflows, leading to automated imaging and analysis setups [5]. Despite their capabilities, the improved performance of modern computers and the increasing abilities of artificial neural networks led to further improvement of low-cost analysis workflows. Especially instance segmentation neural networks such as Mask R-CNN [6] and R-CNN-based neural networks have already demonstrated their usability for the analysis of Petri dish images [7] and microbial localization workflows [8]. In addition to a variety of approaches based on artificial neural networks, there are also methodologies based on heuristic optimization that show similar approaches by using evolution strategies to improve neural network results and drawbacks. For example, there is a fully adaptive alternative for single cell detection [9] without needing labeled images. Due to the required need for datasets for the creation of such neural networks, there are open source datasets of Petri dish images sourced from different imaging setups [8] as well as approaches using style transfer for an artificial extension of the available data [10]. Our proposed approach was implemented to overcome and reduce the challenges of newly developed engraved Petri dishes (SpheroWell), such as manual labeling and analysis of their densely populated surface with closely neighbored objects, where neural networks show insufficient results. The SpheroWell approach adds an engraved grid structure to the Petri dish, requiring grid extraction to allow subsequent analyses. In addition to these already available algorithms, our workflow combines the capabilities of object detection neural networks such as Mask R-CNN with an evolution strategy for extracting the grid structure within the Petri dish, providing an overall improved segmentation and classification result. Therefore, our approach represents a perfect addition to the mentioned state-of-the-art methodologies.

3. Materials and Methods

3.1. Dataset Acquisition and Annotation

The dataset used consists of 20 24-bit images of SpheroWell Petri dish cultures (ASC/TERT1) with a dimension of 4608 × 3456 pixels, as shown in Figure 1. Despite the limited amount of images, up to 4000 objects can be found within the images, including variety in position, rotation, intensity distribution, and class distribution, making the image set representative and usable for algorithm development and evaluation. All of these images were created using an Olympus OM-D E-M1 single-lens reflex digital camera with an Olympus M.Zuiko Digital ED 60mm 1:2.8 46 macro objective. The camera was fixed to a macro stand at a distance of approximately 230 mm from the Petri dish. This lens allowed us to place the Petri dish as frame-filling as possible within the image without a loss in sharpness. The corresponding class labels, as well as the object labels (represented as a polygon surrounding the object), are created by domain experts with the use of the SlideRunner labeling tool [11].

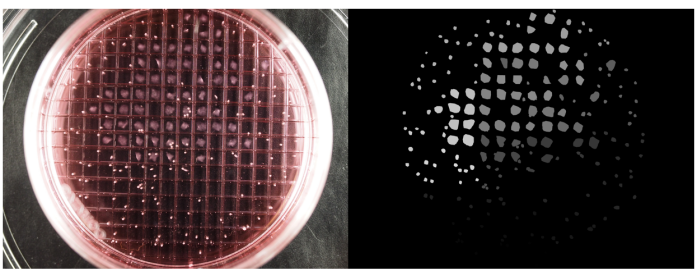

Figure 1. Example of a SpheroWell Petri dish image (**left**) with the individual states of forming spheroids as well as the corresponding object labels on the (**right**). Each labeled object is drawn by a polygon, and its intensity value represents a unique identification number. This number is used for additional data files, including the corresponding class label for each identification number.

3.2. Object Detection Using Deep Learning Techniques

In addition to the need for accurate segmentation of the objects within the Petri dishes, our workflow must also provide each object's position and class (monolayer, forming spheroids, and spheroids). A perfect match for this task is the state-of-the-art instance segmentation method called Mask R-CNN [6]. This segmentation method is based on the original Faster R-CNN [12], which is capable of detecting individual object instances within an image with a class prediction. Additionally, Mask R-CNN provides a binary segmentation mask for each segmented object; this binary mask, in combination with the corresponding classification results, is crucial for our analysis of the individual cell states. In addition to the fast R-CNN and Mask R-CNN predictors of pytorch, for the final Mask R-CNN model, a pretrained resnet50fpn backbone was used.

Based on previous work in this field [13], we included a Unet-based [14] segmentation approach called Unet++ [15] using a pre-trained RegNetY120 backbone out of the segmentation model Pytorch framework [16]. In the previous publication, this semantic segmentation algorithm provided not only a binary segmentation mask for each needed class (cell states), but also an approximation for an instance segmentation based on the connected components labeling result.

Due to the property of the early spheroid stages not being able to overlay each other, we wanted to detect these stages using an xor approach, which was realized using a multiclass Unet++ with a softmax activation function applied within the final network layer, or the Mask R-CNN. In contrast to the early stages, the final spheroids are able to overlay other objects. These spheroids are detected by an additional binary classification, Unet++, realized using a sigmoid activation function within the final layer. The combined segmentation of these two neural networks used in this publication (one for the monolayer and forming spheroid class and one for the spheroids themselves) outperformed all of our approaches using a single multiclass Unet++ for the classification of all spheroid stages.

Both model architectures were trained using the Pytorch lightning framework [17] under Ubuntu 22.04, including data augmentation techniques provided by the albumentations framework [18] for an artificial extension of the available image data. For additional processing steps scikit-learn was used [19]. These augmentation techniques included horizontal/vertical flip, rotation, Gaussian blurring, random shifts of the individual color channels, the brightness and contrast of the whole image, and color saturation. In addition to these data augmentation methods applied to the individual images, we also included image cropping for an extension of our dataset by cropping smaller parts out of the initial image. For final evaluations, subimages with the size of 768 $\times$ 768 pixels were used. Using the combination of data augmentation and image cropping, we artificially extended our training dataset by a factor of 25.

All our models were trained for a maximum of 2000 epochs, selecting the best validation model out of these epochs as the final model used for our analysis, where the data splitting is shown in Figure 2. Both architectures (Mask R-CNN, Unet++) were also trained using two different types of input. On the one hand, we trained our models using the provided raw RGB images sourced from our imaging setup. In addition to extensive evaluation and the potential for improved results, we also used the LAB colorspace, converting these images to their corresponding luminance representation using the LAB color space. This has already shown improved results compared with RGB colorspace in other biomedical applications [20]. Using the luminance information as a standalone image, it is possible to reduce the impact of color shifts between the images due to slight changes in the imaging setup.

During the application of our models, we split the input image into crops of 768 $\times$ 768 pixels, which are segmented one after another until the whole image got processed by our model. For a reduction of artifacts, especially at the borders of each subregion, we repeated this segmentation three times using offsets of the input image so that the borders of the resulting subregions overlap from one segmentation result to another. The final

segmentation result is calculated using the mean of these segmentation masks and the median of the classification mask.

The selection of the used models should only be considered as a representative sample of the already well-established state-of-the-art methods for semantic segmentation as well as instance segmentation.

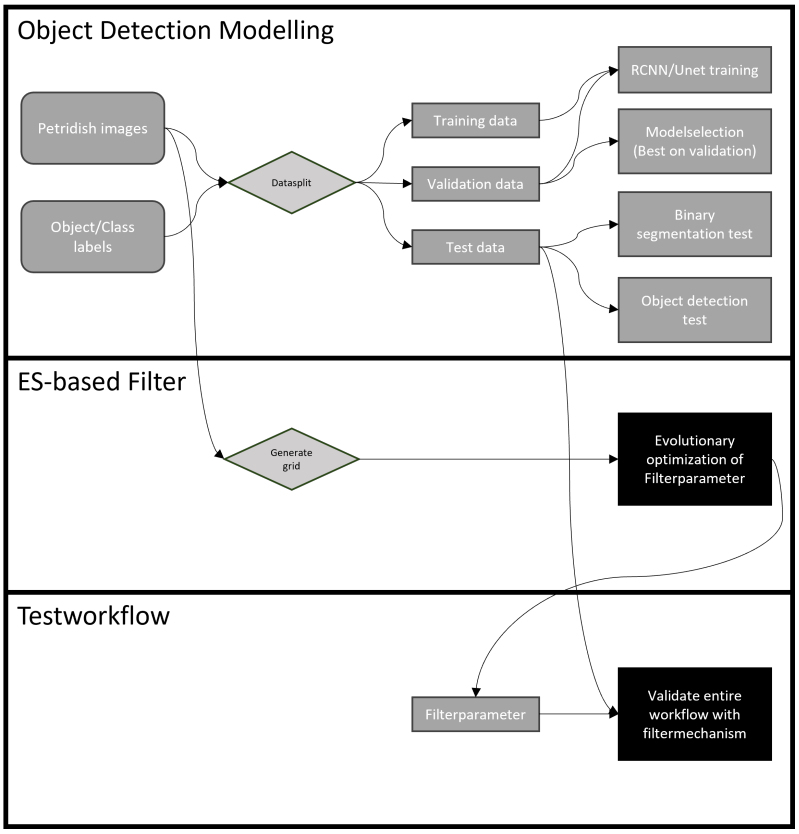

Figure 2. We used our dataset of 20 images of Petri dishes, along with class and object labels, to create and validate individual workflow modules. For the creation of our segmentation/instance segmentation models, we split the initial images into three datasets (training, validation, and testing). Using the training and validation split, we created our Mask R-CNN and our Unet++. Using our validation dataset, we selected the best model for both architectures based on the validation loss. The final testing of our model was split into two parts (binary segmentation and object detection), both using the test dataset. Solely using the Petri dish images without class labels, our evolutionary grid optimization created a grid for each of our images. Based on these grids, we created and tested our parameters for the grid-based filtering. The final testing of our workflow was also based on the separate test dataset.

3.3. Evolutionary Grid Optimization and Computer Vision Techniques for Petri Dish Detection

The grid extraction and mapping are crucial for the overall workflow. Grid mapping allows the extraction and analysis of each single compartment cell within the Petri dish. But more importantly, it improves the overall accuracy of the object localization by using the grid for filtering the segmentation mask predicted by the model in the post-processing step. The mapping itself is done by an evolution strategy, fitting a candidate representing a grid into the original image. The evolution strategy is required due to the insufficient accuracy of more traditional extraction methods like thresholding or Hough Line extraction (shown in Figure 3) [21]. To optimize a candidate, a fitness function is applied to calculate the difference between the candidate's solution, and a preprocessed image is required for the error calculation. The required steps are shown in Figure 3.

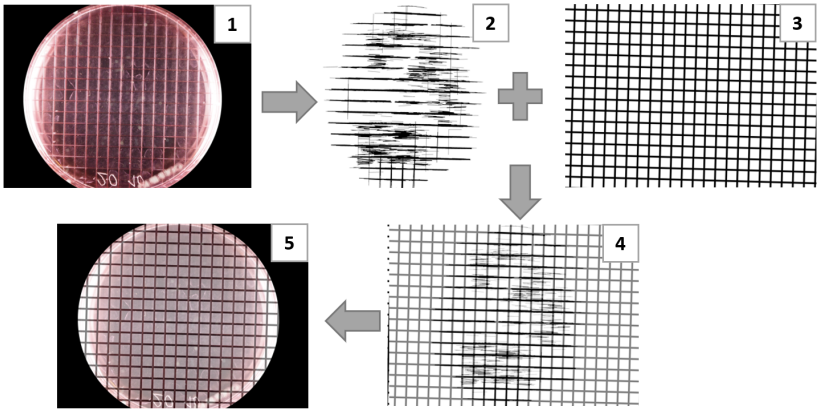

Figure 3. Overview of steps for the evolution strategy. 1: Extracted Petri dish with Hough Circles (insufficient extraction of grid cells), 2: Extracted Hough Lines 3: grid of a solution candidate, 4: Compute fitness of solution candidates with overlap based on Dice coefficient, 5: Final results.

3.3.1. Petri Dish Preprocessing for Fitness Function Evaluation

Image processing is required to extract the underlying pattern as a binary mask of the grid, which allows a fitness calculation of an error between the preprocessed image and a candidate. This error is defined by the inverted Dice coefficient between the candidate's estimated grid and the preprocessed image's actual grid structure [22]. Multiple preprocessing steps are performed to create the required binary mask. First, Hough Circle extraction is applied to detect the Petri dish itself [23]. Second, a filter mechanism identifies parts of the actual grid lines that are further used for Hough Line detection to estimate the entire grid in the image. Multiple parameter values and parameter sets were compared to validate the used parameters and evaluate the robustness of the workflow. The evaluations for each preprocessing step are based on the fitness function. The final extracted grid is additionally evaluated based on a ground truth grid and the Dice coefficient.

The extraction of the Petri dish is done by detecting the dish's circular outer border with the Hough Circle algorithm. After blurring the image for improved edge detection, a stable threshold for the Canny edge detection, accumulator threshold, and suitable minimum/maximum radius sizes are determined manually based on unseen images used only for training and parameter evaluation (Canny threshold: 125, accumulator threshold: 200, radius (min): 37.5%, radius (max): 75%). The size of the image itself defines the radius size to stay independent for different cameras, allowing better adaptability. Due to the thickness of the border of the Petri dish, multiple Hough Circles are detected and filtered by defining a minimum circle distance, eliminating neighboring circles. To further increase robustness, based on the ratio of the image shape and center, circles outside of a predefined area are discarded. An evaluation of metaparameters was applied to show the applicability of multiple parameter sets and to extract the best metaparameters. Multiple settings for the Hough Circles Canny edge detection thresholds were used, and the mentioned minimum and maximum radius were evaluated. Finally, if multiple fitting circles are proposed by the algorithm, the best-fitting circle is chosen based on the highest sum of intensity covered by the proposed circle region. A similar approach used in previous research provides reasonable results also on different resolutions and positions of the Petri dish within the image, showing its robustness and applicability on a slightly changed setup [13].

For the extraction of the grid structure, it is necessary to extract as many lines as possible while reducing noise, false positives, and artifacts. Therefore, the lines are detected using a combination of edge and line detection with Hough Lines. First, intensity equalization is applied with CLAHE to increase the contrast and emphasize the brighter lines (and their edges) of the Petri dish in the image. For the detection of the lines, edge detection is applied. Due to the thickness of the grid lines, multiple edges are detected for a single grid line, which are merged into a single thicker line using closing as morphological operations.

Hough Line detection is applied to extract lines only based on the extracted edges of cells within the compartment. The minimum length and maximum gap are a priori defined for the Hough Line extraction to allow a robust line extraction parameter like an accumulator threshold. All parameters are derived based on ratios compared with the overall size of the taken image to allow size-independent execution of the line extraction. The parameters are derived manually from a set of training images (accumulator threshold: 50, minimum length: 400 px (10%), maximum gap: 50 px (1.5%). After the Hough Line extraction, closing is used to connect neighboring lines. Additionally, based on an angle that does not appear in real-world scenarios, wrongly detected lines that are not part of the grid are identified as noise and removed. The maximum and minimum angles are defined as $10/-10$ degrees to allow the capture of a simple picture of the Petri dish without perfect alignment of all utilities. Our experiments further showed a potential benefit in using two different values for vertical and horizontal lines depending on the direction of the light source, which highlights only vertical or horizontal lines.

The final extracted binary image contains valid lines of the grid structure in the original image to be used for the fitness function of the evolution strategy. The extracted lines themselves cannot be used to extract the whole grid. Reflections and shadows in the original image reduce the visibility of all lines, and therefore, not all lines could be extracted sufficiently with this approach. The extracted binary image still contains the overall grid structure of the underlying engravings in the Petri dish that can be used as a template to fit an artificially created grid (solution candidate) into the binary mask.

For the line detection, multiple parameters are necessary for a successful detection, similar to the circle detection. Besides filtering algorithms used to remove non-vertical or non-horizontal lines, multiple parameters are used for the line detection itself. For the workflow at hand, the most important parameters are the accumulator threshold, the maximum gap, and the minimum length of lines. These also have been evaluated.

The above methods effectively extract sufficient horizontal and vertical lines from the Petri dish. This is of utmost importance as it serves as a basis for accurately evaluating possible solution candidates of the evolution strategy algorithm. Furthermore, this process ensures that the assessment is carried out with high precision and accuracy.

3.3.2. Solution Candidate Definition and Grid Optimization

Due to missing or incomplete grid lines, shown in Figure 3 in the extracted binary mask, the grid has to be extrapolated or estimated. Therefore, an evolution strategy is used to approximate the entire grid. As long as the underlying structure of the grid is visible within the extracted binary mask (a minimum of 50% vertical and horizontal lines), this mask can be used for highly accurate grid estimations using our evolution strategy. The overall idea is to use solution candidates, which represent the parameters used in grid reconstruction. The solution candidate grid is evaluated through the binary image used in the fitness function and adapted in the strategy mutation step.

A solution candidate represents a grid with horizontal and vertical lines over the whole image defined by a certain distance between each line (shown in Figure 3 sub-image 3). Each solution candidate results in the grid estimation of the grid structure within the Petri dish. Each solution candidate is, therefore, defined by four mutable parameters and one fixed parameter spanning the estimated grid image. This fixed parameter is the width of the line drawn into the estimated image to span the grid. The four mutable parameters are loosely based on affine transformation parameters, which allow the manipulation of the grid while preserving the underlying structure. Our grid candidates are defined by the following parameters:

- Rotation: rotation of the image to a horizontal line.
- Scaling: width of an engraved grid cell (without borders).
- Vertical offset: vertical translation of the grid regarding the top image border.
- Horizontal offset: horizontal translation of the grid regarding the left image border.

With these parameters, the candidates are generated. The initial configuration of these parameters is randomly set within a certain range of valid values. Valid values are defined by a maximum rotation (4 degrees), a minimum distance between lines (150 pixels), and a predefined grid thickness (31 pixels). These values can be changed for different setups or alternatively ignored, resulting in slower convergence.

The fitness of each candidate is defined as the inverted Dice coefficient calculated between the solution candidate grid image (sub-image 3 in Figure 3) and the image with the extracted grid lines represented by Hough Lines in the extracted binary image (sub-image 2 in Figure 3). The used formula for the fitness function defined by the dice coefficient is shown in Equation (1). The calculated error given by the Dice coefficient between the predicted grid of the solution candidate and the extracted lines of the preprocessed petri dish image directly translates to mismatched pixels of unmatched lines. By adapting the grid through mutation operations in the solution candidate, an error reduction can be achieved over multiple generations by increasing the overlap between the lines in the extracted binary mask and the estimated grid of the solution candidate. Due to the repetitive pattern, a gradual improvement of the estimated grid is possible until the candidate's fitness can no longer be improved. For the adaptation of sigma and the generation of a new population, the one-fifth rule was used.

Two approaches were used for the evaluation of the grid prediction results. Important metaparameters for the ES are population size, starting σ, and maximum iterations needed. First, different metaparameters of the evolution strategy were evaluated based on the fitness score to evaluate the most important parameters and best parameter set. Second, the predicted grids with their corresponding metaparameter sets were evaluated based on a handcrafted ground truth image

$$FitnessFunction = 1 - \frac{2 * TP}{2 * TP + FP + FN} \quad (1)$$

where Dice coefficient is defined by TP: true positive; FP: false positive; and FN: false negative.

The candidate parameters are used to draw the estimated grid into a blank image. As a solution, the candidate describes all lines of a grid, contrary to the extracted binary image, which misses some parts of grid lines; a perfect error of 0 is not achievable. Still, the best estimation results in the lowest amount of error, representing the best approximation for the grid structure within the Petri dish.

The reason for being unable to reach an error of 0 depends on the underlying image. Due to lighting conditions, etc., not all lines can be extracted perfectly by the Hough Lines. Additionally, artifacts can still be included in the extracted Hough Line binary image. These factors make a fitness with a dice coefficient lower than 0 unlikely. However, since the lines of the grid are the most dominant and repetitive pattern within the image, a lower error still represents a better solution. Depending on the image, fitness values between 0.02 and 0.1 have to be found as sufficient results, whereas in most images, a fitness value larger than 0.2 is considered insufficient and often represents the fitness value of the starting candidate.

For a reduction of the time until the fitness converges and the risk of insufficient local optima, the search space is reduced based on problem-specific constraints. A minimum and maximum width for the grid is defined since the resulting grid always takes up a specific part of the image based on the distance the picture was taken. The size of a grid cell is currently defined as 10% of the image's height. At the same time, this value is also used for the maximum offset. We also defined a minimum width for the grid cells to prevent the overuse of grid lines within an image. If a mutation would over-/undershoot the hard constraints, the solution candidate is recalculated until parameters are within the constraints. These constraints can easily be adapted to allow different engraved grids or repetitive patterns in the future.

3.4. Spheroid Classification Workflow

We created our classification workflow using separate modules for object detection and grid optimization, as shown in Figure 4. For the tested segmentation models (Unet++ or Mask R-CNN), the input RGB image is used in its original form or converted into its luminance representation for the detection of the individual objects within the Petri dish. Independent of the segmentation, our workflow predicts the grid lines based on the detected grid within the Petri dish. Based on the resulting segmentation mask, our workflow uses the detected grid within the Petri dish to individually analyze each grid cell. Since it is not possible to find multiple valid objects within one cell, except finalized spheroids, our classification focuses on the monolayer and forming spheroid stages. For each detected object (monolayer and forming a spheroid), the calculated confidence score provided by the neural network is used to make the decision on which class should be applied for the entire cell. Since mirroring and cut-offs of the Petri dish at the border of the dish may negatively influence the grid cell detection, our workflow weights scores near the cell center higher than border objects using a Gaussian distribution into account. Additionally, a second weight takes the relative size of the detected object compared with the combined size of all detected objects within the grid cell into account. Both weights lead to a preference for large connected objects that reach the center of the grid cell. Based on the resulting score, the highest-scoring object's class is used to classify each detected object within the cell. This approach also leads to a correct size measurement of the whole grid cell objects, which would otherwise be wrongly detected as multiple objects due to insufficient separation of segmented cells since neural networks often fail to separate close or overlapping objects. This grid cell-based class correction and filtering leads to the filtering of all wrongly detected objects outside the Petri dish, as well as invalid reflections of objects occurring on the borders of the Petri dish. After each grid cell is processed, the resulting segmentation mask is combined with the unfiltered and unprocessed mask of finalized spheroid objects.

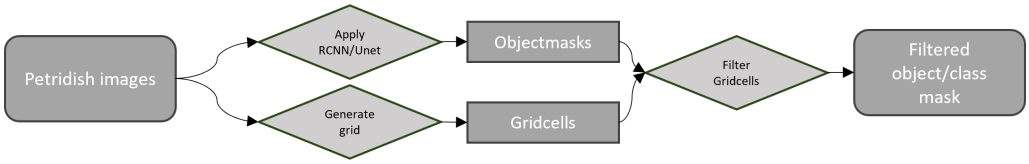

Figure 4. Simplified representation of the whole classification workflow, starting with the image of a Petri dish carrying the individual objects. Based on this image, our workflow generates a grid using the evolution strategy-based optimization process and provides the position of each grid cell, as well as a classification mask of all individual objects determined by our neural network. Both of these results are fed into the grid filtering mechanism, which processes the classification results based on the grid cells and outputs the final classification result for each object.

4. Results

Figure 5 shows the applicability of our evolution strategy sub-workflow. Due to the noise of additional lines of the preprocessing the global optimum cannot result in a fitness score of 0.0. It also shows that the evolution strategy is robust to noise and wrong Hough Lines, which do not represent any lines of the original and correct engraved grid of the RGB image. It clearly shows that these results can be used further for post-processing and correct grid cell extraction to improve overall accuracy further.

Regarding evaluating the metaparameters used for the circle detection, none of the results show large differences in the final average fitness. Therefore, small parameter changes have no significant effect on the final results of the circle detection. This also aligns with the expected results since only small parts of single-cell compartments on the borders could potentially be missed due to a slight misalignment of the resulting circle. However, it shows a stable circle detection when using different parameters. Additionally, the maximum radius between 0.7 and 0.8 and the minimum radius between 0.3 and 0.4

provide stable results on average. A slight variance with a maximum error difference of 0.008 in the fitness score can be observed in the evaluated parameters. A robust parameter combination for circle detection was used with a Canny edge threshold of 125 and a threshold for center detection of 200 in combination with 0.375 for the minimum and maximum 0.75 circle radius.

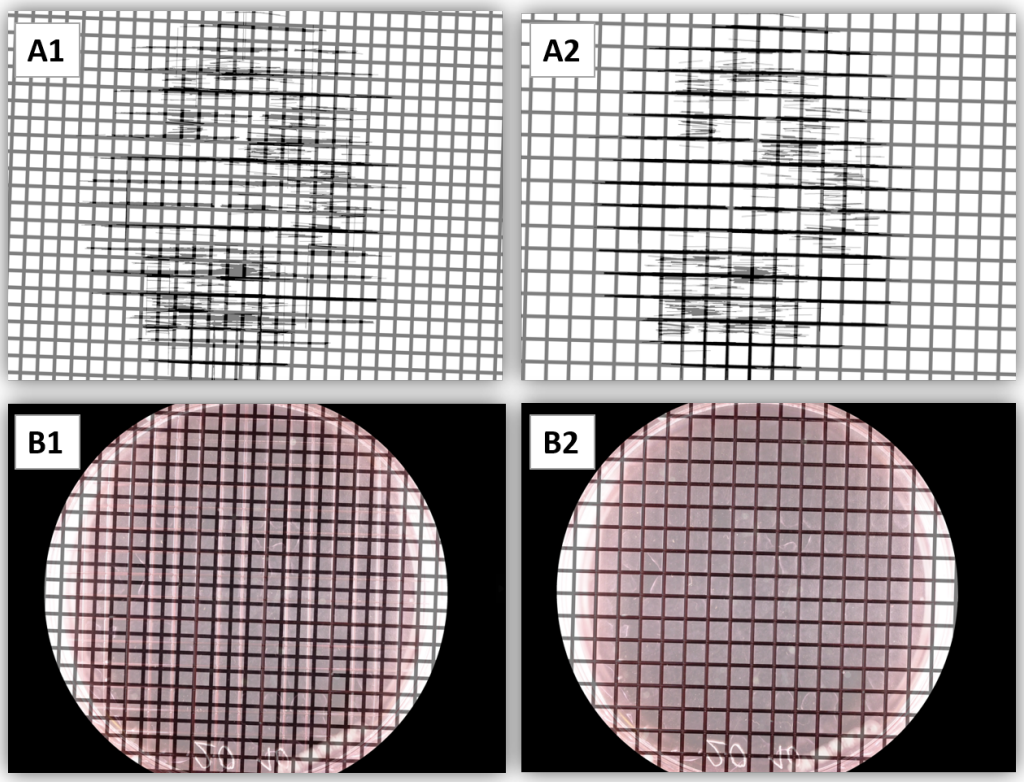

Figure 5. Comparison of first epoch (**A1**,**B1**) and last epoch (**A2**,**B2**) of an evolution strategy run. Images A1 and A2 show a grid represented by a solution candidate, and images (**B1**,**B2**) show the final solution on the RGB image with the extracted Petri dish. The result clearly shows the improvement resulting in an almost perfectly aligned grid (fitness: 0.0844). Additionally, **A** and **B** show the robustness of the evolution strategy to noise and wrong lines of the preprocessing.

For the line detection, a difference in the importance of certain parameters and the best parameter set could be obtained and used (shown in Figure 6). No difference in average fitness scores and a high variance can be observed when changing the accumulator threshold of the line detection. Only a small difference in average values while having a huge range of fitness scores can be observed in the minimum length. This suggests another parameter influences the results strongly. This can be observed in the maximum gap, showing a strong correlation between using smaller gaps and better results. Higher scores by using smaller gaps could be explained by the minimization of noise within the image, taking only longer lines with small gaps into account representing the grid lines. The best results are achieved with 70 for accumulator threshold, 30 for maximum gap size, and 300 for minimum length.

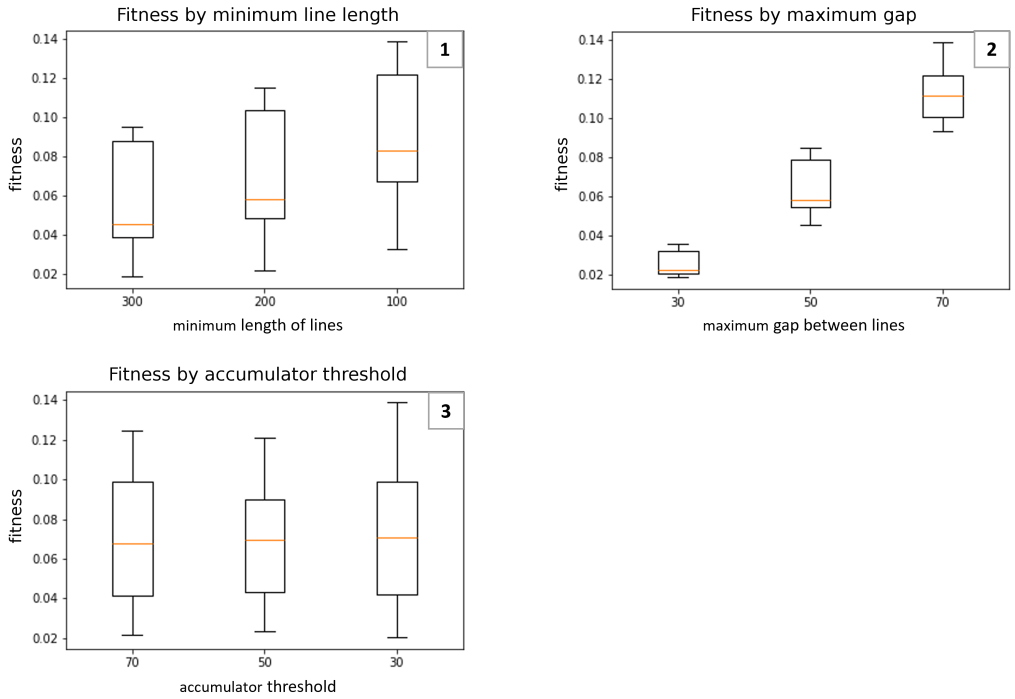

Figure 6. Comparison of validated parameters and corresponding values used for the line extraction process by fitness value (lowest is best). Showing the influence and robustness of best and most important parameters and their corresponding values. Large differences in the minimum gap parameter can be observed.

4.1. Evolution Strategy Metaparameter Results

The results suggest large differences in the metaparameters of the evolution strategy, showing the importance of choosing the correct parameters to achieve good fitness (shown in Figure 7). Especially the population size (λ, μ) affects the result. Therefore, a population using 40 for λ and 60 for μ does provide the best results; as an alternative, a population size of (20, 40) could be used. Only slight differences can be observed in the maximum iterations and starting σ, especially considering the high variance (mostly explained by the population size). With these results, the used parameter for the evolution strategy is (40, 60) for population, 40 for iterations, and 0.05 as starting σ. These settings achieved the overall third-best result and best results with a maximum iteration of 40. Due to its quick computation speed, it was chosen as the final evaluation. When combined with the best fitting line parameters, the best achieved results reach a fitness of 0.0169.

Additionally, to evaluate the results of the fitness, the estimated grid was evaluated based on a ground truth grid. The final score is based on the Dice coefficient between the predicted and ground truth images. Overall, the fitness score and parameter evaluations show similar results. Again, (40, 60) for population shows the best overall results with a Dice coefficient of the grid of 0.6768 compared with bad parameter sets with a Dice coefficient of 0.3173. As observed in the results of Figure 8, the outcome of the ground truth evaluation follows the same pattern as the fitness evaluation in regard to the overall performance between parameter settings.

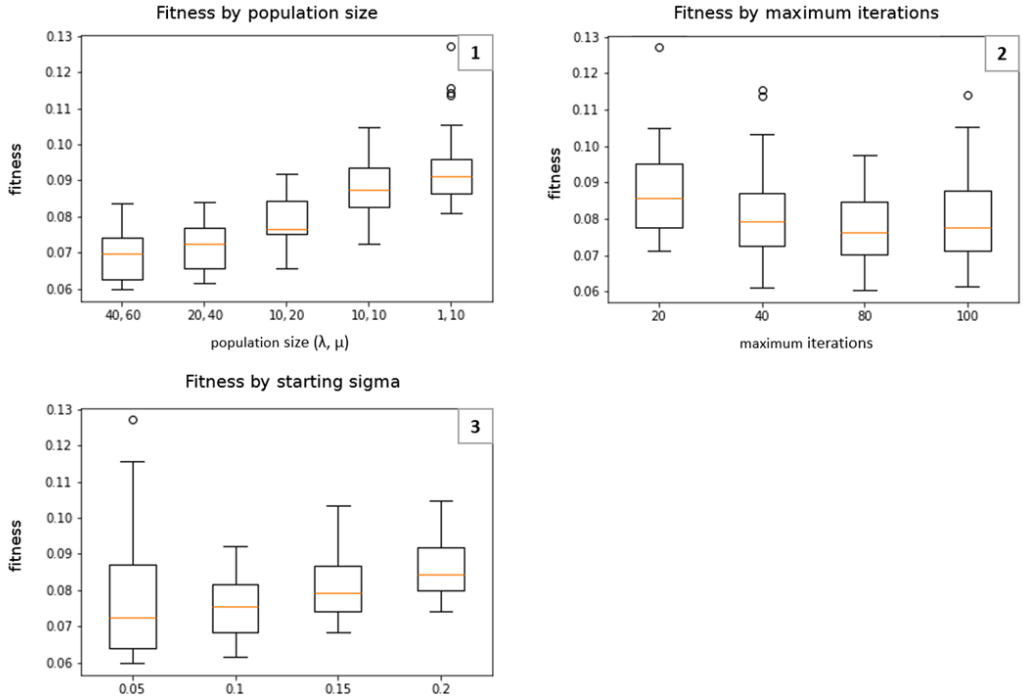

Figure 7. Comparison of validated parameters and corresponding values used for the ES by fitness value (lowest is best). Showing the influence and robustness of best and most important parameters and their corresponding values. Large differences in population size can be observed.

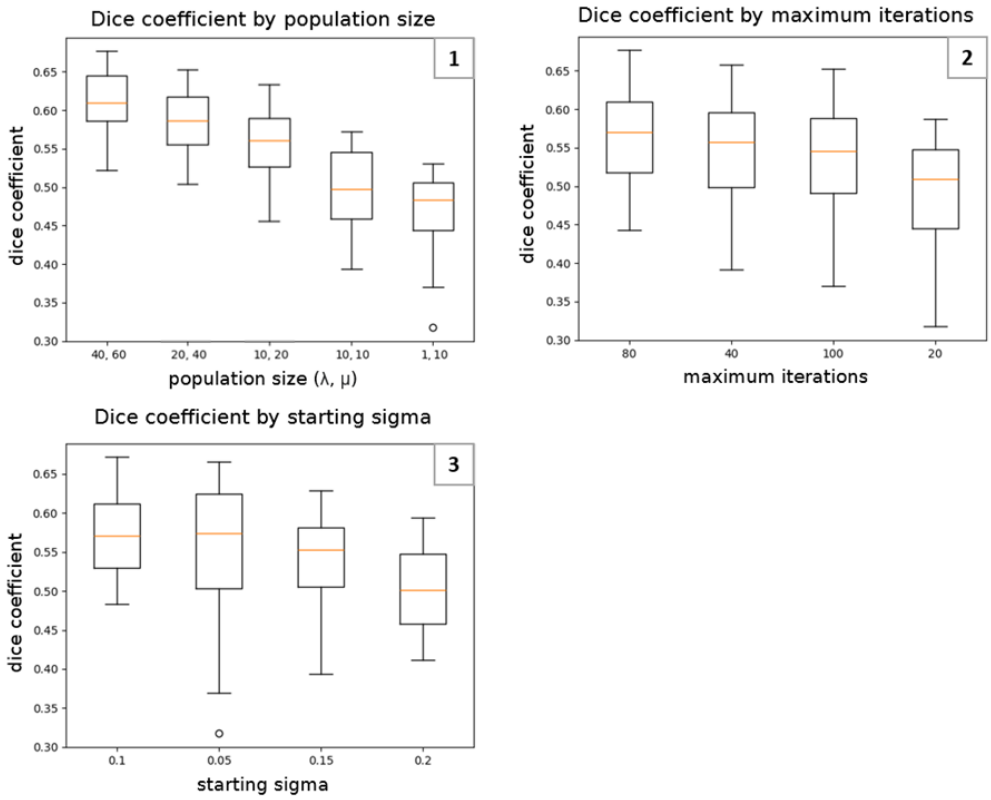

Figure 8. Comparison of validated parameter values used for the ES by Dice coefficient (highest is best) based on a ground truth image for detected lines and corresponding. Showing the influence and robustness of best and most important parameters and their corresponding values. Large differences in population size can be observed.

4.2. Neural Network Application Results

For an accurate and reproducible comparison of our evolution strategy-based post-processing compared with the raw segmentation results, we first selected the best segmentation algorithm out of our 4 created models.

As shown in Table 1, representing the achieved segmentation results using the validation dataset, it is the case that the conversion of RGB images to their corresponding luminance variant leads to improved segmentation quality for both types of neural networks. Comparing the segmentation quality of the used Mask R-CNN and Unet++, it is also the case that the Mask R-CNN outperforms the Unet++.

Table 1. Comparison of binary segmentation results on the validation dataset between the Mask R-CNN and the Unet++.

Model	Luminance Conversion	Dice	Jaccard
Mask R-CNN	yes	0.8982	0.8154
Unet++	yes	0.8910	0.8036
Mask R-CNN	no	0.8970	0.8134
Unet++	no	0.8785	0.7838

Our decision to use Mask R-CNN based on luminance input data is also supported by the results achieved using the test dataset as shown in Table 2. Based on this model selection, we analyze the classification performance of our luminance-based models. As shown in Table 3, it is still the case that the Mask R-CNN trained on the luminance data provides the highest accuracy. In addition, it can also be seen that the use of our grid-based filter improves the classification accuracy, especially the accuracy of the monolayer class.

Table 2. Comparison of binary segmentation results on the test dataset between the Mask R-CNN and the Unet++.

Model	Luminance Conversion	Dice	Jaccard
Mask R-CNN	yes	0.9069	0.8298
Unet++	yes	0.9039	0.8249
Mask R-CNN	no	0.9014	0.8206
Unet++	no	0.8974	0.8141

Table 3. Comparison of classification and localization results on the test dataset between the Mask R-CNN and the Unet++ based on the luminance-only data compared with the non-filtered version.

Model	ES Based Gridfilter	F1 Monolayer	F1 Forming Stage	F1 Final Spheroid
Mask R-CNN	yes	0.7309	0.8562	0.8829
Unet++	yes	0.4	0.8135	0.8127
Mask R-CNN	no	0.2879	0.8187	0.8829
Unet++	no	0.0986	0.6615	0.8127

Further analyzing the classification accuracy of the best performing model, Table 4 provides detailed classification measures for our Mask R-CNN trained on luminance-only images with and without the use of our grid-based filter.

In particular, the significant improvement of the precision score of the monolayer class highlights the improved ability to detect this class correctly. The low precision and still high recall score of the raw, unfiltered result shows the model's incapability to correctly separate the monolayers from the similar-looking empty Petri dish bottom without our grid-filtering-based post-processing, showing the importance of the filtering process based on the grid.

Table 4. Classification and localization measures over all classes on the test dataset based on the best performing model represented by the luminace Mask R-CNN.

Class	ES Based Gridfilter	F1	Precision	Recall
Monolayer	yes	0.7309	0.8477	0.6423
Monolayer	no	0.2879	0.1802	0.7154
Forming stage	yes	0.8562	0.7903	0.9343
Forming stage	no	0.8187	0.7115	0.9639
Final spheroid	yes	0.8829	0.8725	0.8935
Final spheroid	no	0.8829	0.8725	0.8935

Even if the difficult-to-classify monolayer class is still not perfectly detected using our post-processing workflow, the example images of our test dataset shown in Figure 9 prove the significant quality improvement due to the corrected classification by our post-processing workflow.

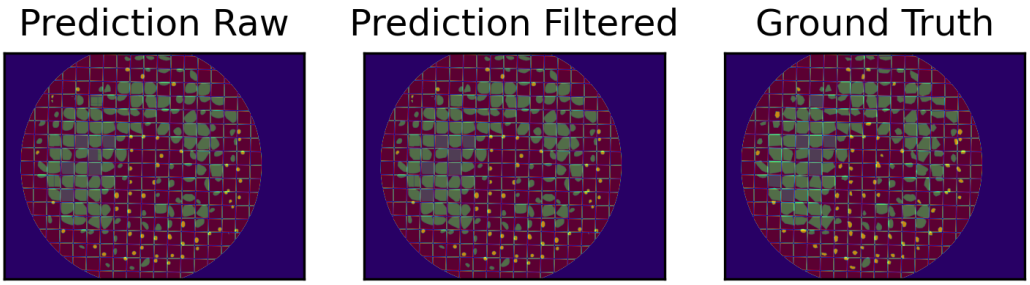

Figure 9. Comparison of the raw classification result and the filtered/corrected version in comparison with the ground truth. (blue: monolayer, green: forming state, yellow: final spheroid).

5. Discussion

Our workflow provides an additional approach for post-processing segmentation results used for object detection workflows. Our results clearly highlight the benefit of including additional features in the form of grid structures in the final classification process that were not part of the training of the machine learning model. The use of an evolution strategy as the base for the detection of the grid structure within Petri dish images provides an adaptive approach for different types of imaging setups and Petri dishes. Future work should also include parameters for different types of culture dishes without the requirement that individual grid cells have to provide a quadratic shape.

Limited by the small amount of data, the presented work could also benefit from the use of the mentioned data generation methodology mentioned within the state-of-the-art section. Still, we decided to exclude this methodology for our work to inhibit any bias during the result analysis that may be included due to the use of this methodology. We believe that a combination of our workflow with methods like data generation could lead to an improvement of the final classification due to the lack of open source datasets providing the required grid structure within the Petri dishes.

Our selection of segmentation neural networks should be considered valid examples without the goal of representing the best neural network architectures for this type of use-case. Our intention during the model selection was to include state-of-the-art and well-established segmentation neural network architectures for cell segmentation tasks for semantic segmentation (Unet++) as well as instance segmentation (Mask R-CNN) to highlight the flexibility of our post-processing workflow, being able to be combined with different types of neural networks. Still, we want to recommend the use of Mask R-CNN-based neural networks for this kind of workflow, which is also supported by the literature mentioned within the state of the art section.

The results achieved using our workflow highlight the need for new post-processing algorithms for the use case of deep learning-based object detection. Despite the high

accuracy of such neural networks, it is still the case that error-correcting post-processing workflows like the presented grid-based approach could significantly improve the over-all classification accuracy. As shown by the comparison of the unfiltered and filtered results for the monolayer class, it is also possible to compensate the classification weaknesses for a class that seems to be exceptionally challenging to separate from the background.

Author Contributions: Conceptualization, J.S. (Jonas Schurr), H.J. and A.H.; Methodology, J.S. (Jonas Schurr), H.J., S.W. and A.H.; Validation, J.S. (Jonas Schurr) and H.J.; Formal analysis, J.S. (Jonas Schurr), H.J. and A.H.; Resources, M.F.; Data curation, M.F. and S.N.; Writing—original draft, J.S. (Jonas Schurr) and H.J.; Writing—review & editing, J.S. (Jonas Schurr), H.J., M.F., J.S. (Josef Scharinger), S.W. and S.N.; Visualization, J.S. (Jonas Schurr) and H.J.; Supervision, J.S. (Josef Scharinger), S.W. and S.N.; Project administration, S.W. and S.N.; Funding acquisition, S.N. and S.W. All authors have read and agreed to the published version of the manuscript.

Funding: The research presented in this paper was backed by the Center of Excellence for Technical Innovation in Medicine (TIMED), the Dissertation Program of the University of Applied Sciences Upper Austria, and the Austrian Research Promotion Agency (FFG), under project number 881547 and Industrienahe Dissertation number 867720.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: The dataset presented in this article is not readily available because the data is part of an ongoing study. Requests to access the dataset should be directed to the corresponding author (Jonas Schurr).

Conflicts of Interest: The authors declare no conflicts of interest.

References

1. Johnstone, B.; Hering, T.; Caplan, A.; Goldberg, V.; Yoo, J. In vitrochondrogenesis of bone marrow-derived mesenchymal progenitor cells. *Exp. Cell Res.* **1998**, *238*, 265–272. [CrossRef]
2. Zhang, L.; Su, P.; Xu, C.; Yang, J.; Yu, W.; Huang, D. Chondrogenic differentiation of human mesenchymal stem cells: A comparison between micromass and pellet culture systems. *Biotechnol. Lett.* **2010**, *32*, 1339–1346. [CrossRef]
3. Fürsatz, M.; Gerges, P.; Wolbank, S.; Nürnberger, S. Spherowell, a novel system inducing autonomous spheroid formation of cell monolayers. *Osteoarthr. Cartil.* **2021**, *29*, S407–S408. [CrossRef]
4. Schneider, C.A.; Rasb, W.S.; Eliceiri, K.W. NIH Image to ImageJ: 25 years of image analysis. *Nat. Methods* **2012**, *9*, 671–675. [CrossRef]
5. Ding, X.; Vogel, M.; Boschke, E.; Bley, T.; Lenk, F. PetriJet Platform Technology: An Automated Platform for Culture Dish Handling and Monitoring of the Contents. *J. Lab. Autom.* **2015**, *20*, 447–456. [CrossRef]
6. He, K.; Gkioxari, G.; Dollár, P.; Girshick, R. Mask R-CNN. In Proceedings of the 2017 IEEE International Conference On Computer Vision (ICCV), Venice, Italy, 22–29 October 2017; pp. 2980–2988.
7. Yang, F.; Zhong, Y.; Yang, H.; Wan, Y.; Hu, Z.; Peng, S. Microbial Colony Detection Based on Deep Learning. *Appl. Sci.* **2023**, *13*, 10568. [CrossRef]
8. Majchrowska, S.; Pawlowski, J.; Gula, G.; Bonus, T.; Hanas, A.; Loch, A.; Pawlak, A.; Roszkowiak, J.; Golan, T.; Drulis-Kawa, Z. AGAR a microbial colony dataset for deep learning detection. *arXiv* **2021**, arXiv:2108.01234.
9. Haghofer, A.; Dorl, S.; Oszwald, A.; Breuss, J.; Jacak, J.; Winkler, S. Evolutionary optimization of image processing for cell detection in microscopy images. *Soft Comput.* **2020**, *24*, 17847–17862. [CrossRef]
10. Pawłowski, J.; Majchrowska, S.; Golan, T. Generation of microbial colonies dataset with deep learning style transfer. *Sci. Rep.* **2022**, *12*, 5212. [CrossRef]
11. Aubreville, M.; Bertram, C.; Klopfleisch, R.; Maier, A. SlideRunner—A Tool for Massive Cell Annotations in Whole Slide Images. In *Bildverarbeitung für die Medizin 2018*; Springer: Berlin/Heidelberg, Germany , 2018; pp. 309–314.
12. Ren, S.; He, K.; Girshick, R.; Sun, J. Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks. In Proceedings of the Advances In Neural Information Processing Systems (NIPS), Montreal, QC, Canada, 7–12 December 2015.
13. Schurr, J.; Haghofer, A.; Fürsatz, M.; Janout, H.; Nürnberger, S.; Winkler, S. In Vitro Quantification of Cellular Spheroids in Patterned Petri Dishes. In Proceedings of the BIOSTEC 2023, Lisbon, Portugal, 16–18 February 2023. [CrossRef]
14. Ronneberger, O.; Fischer, P.; Brox, T. U-net: Convolutional networks for biomedical image segmentation. In *Lecture Notes in Computer Science (Including Subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*; Springer: Berlin/Heidelberg, Germany, 2015.

15. Zhou, Z.; Rahman Siddiquee, M.; Tajbakhsh, N.; Liang, J. Unet++: A nested u-net architecture for medical image segmentation. In *Deep Learning in Medical Image Analysis and Multimodal Learning for Clinical Decision Support: 4th International Workshop, DLMIA 2018, and 8th International Workshop, ML-CDS 2018, Held in Conjunction With MICCAI 2018, Granada, Spain, September 20, 2018, Proceedings 4*; Springer International Publishing: Cham, Switzerland, 2018; pp. 3–11.
16. Iakubovskii, P. Segmentation Models Pytorch. GitHub Repository. 2019. Available online: https://github.com/qubvel/segmentation_models.pytorch (accessed on 23 August 2023).
17. Falcon, W.A. Pytorch Lightning. GitHub. 2019. Available online: https://github.com/PyTorchLightning/pytorch-lightning (accessed on 19 September 2024).
18. Buslaev, A.; Kalinin, A. Albumentations: Fast and flexible image augmentations. *arXiv* **2018**, arXiv:1809.06839. [CrossRef]
19. Walt, S.; Schönberger, J.; Nunez-Iglesias, J.; Boulogne, F.; Warner, J.; Yager, N.; Gouillart, E.; Yu, T. scikit-image: Image processing in Python. *PeerJ* **2014**, *2*, e453. [CrossRef]
20. Haghofer, A.; Ebner, T.; Kainz, P.; Weißensteiner, M.; Ghaffari-Tabrizi-Wizsy, N.; Hatab, I.; Scharinger, J.; Winkler, S. Automated Data Adaptation for the Segmentation of Blood Vessels. In Proceedings of the Biomedical Engineering Systems and Technologies—15th International Joint Conference (BIOSTEC 2022), Online, 9–11 February 2022; Revised Selected Papers; Springer Nature: Cham, Switzerland, 2023; pp. 53–72.
21. Aggarwal, N.; Karl, W. Line detection in images through regularized Hough transform. *IEEE Trans. Image Process.* **2006**, *15*, 582–591. [CrossRef]
22. Dice, L. Measures of the amount of ecologic association between species. *Ecology* **1945**, *26*, 297–302. [CrossRef]
23. Kerbyson, D.; Atherton, T. Circle detection using Hough transform filters. In Proceedings of the 1995 International Conference on Electric Railways in a United Europe, IET, Amsterdam, The Netherlands, 27–30 March 1995.