

Easing the Reuse of ML Solutions by Interactive Clustering-based Autotuning in Scientific Applications

1st Hamideh Hajiabadi

2nd Lennart Hilbert

3rd Anne Koziolak

*Helmholtz Information and Data Science
School for Health (HIDSS4Health)
Karlsruhe Institute of Technology
Karlsruhe, Germany
hamideh.hajiabadi@kit.edu*

*Zoological Institute and Institute
of Biological and Chemical Systems
Karlsruhe Institute of Technology
Karlsruhe, Germany
lennart.hilbert@kit.edu*

*Information Security
and Dependability (KASTEL)
Karlsruhe Institute of Technology
Karlsruhe, Germany
Anne.koziolak@kit.edu*

Abstract—Machine learning techniques have revolutionised scientific software projects. Scientists are continuously looking for novel approaches to production-quality reuse of machine learning solutions and to make them available to other components of the project with satisfactory quality and low costs. However, scientists often have limited knowledge about how to effectively reuse and adjust machine learning solutions in their particular scientific project. One challenge is that many machine learning solutions require parameter tuning based on the input data to achieve satisfactory results, which is difficult and cumbersome for users not familiar with machine learning. Autotuning is the common technique for potentially adjusting the parameters based on the data, but it requires a well-defined objective function to optimize for. Such an objective function is commonly unknown in exploratory scientific research such as biological image segmentation tasks. In this paper, we propose a framework based on the novel combination of autotuning and active learning to ease and partially automate the reuse effort of machine learning solutions for scientists in biological image segmentation cases. Underlying this combination is a mapping between an object type and specific parameters applied during the segmentation process. This mapping is iteratively adjusted by asking users for visual feedback. We then through a biological case study demonstrate that our method enables tuning of the segmentation specifically to object types, while the selective requests of user input reduce the number of user interactions required for this task.

Index Terms—Reuse of Machine Learning Solutions, Interactive biological image segmentation, Auto tuning

I. INTRODUCTION

Artificial intelligence (AI) and machine learning (ML) have the potential to revolutionise scientific software. A particular characteristic of scientific software is that it is commonly developed and/or reused by scientists [1]. However, scientists often have limited knowledge about how to reuse and deploy machine learning solutions effectively. For example, in biological pipelines, ML segmentation solutions often need to be reused and deployed by scientists who are not much familiar with programming. Despite their lack of programming skills, biologists require the target objects to be segmented in a way that is satisfactory for further analysis.

One of the main challenges for scientists is to choose between many ML solutions and adjust (i.e. tune) parameters based on their own data to achieve their objectives with effectiveness, efficiency, and satisfaction in the context of use. This becomes even more challenging in scenarios where multiple types of input data (e.g. different types of objects in image segmentation) are to be handled by an ML solution and where *different* parameter settings would be optimal for each such type. This makes the search for optimal parameters even more difficult, as an improvement for one type of input may deteriorate the prediction quality for another type of input and may lead to cycling back and forth in a manual search. This challenge of selecting ML solutions, deploying them and adjusting their parameters is especially difficult and cumbersome for users not familiar with ML.

In supervised ML approaches, the reuse and deployment can be easier as this tuning can be done automatically by optimising an objective function that compares the predicted label with the true label from the training data. However, the adjustment of these algorithms for unknown data sets poses a major challenge in practical application. One specific hurdle is the annotation of objects by experienced domain specialists, which often forms the basis for the adjustment of segmentation algorithms, but is extremely labor-intensive. In unsupervised ML approaches, where usually there is no training data to define an objective function on, automatically reuse and tuning is only possible if another kind of objective function that assesses the quality of prediction results is available. However, in many use cases, such as image segmentation, no objective function that captures all aspects of quality of solution is available. One approach to tool-supported ML solution selection and parameter tuning is to randomly generate solutions, to let the users assess them and, based on that, to learn over time what makes up a good solution (so-called active learning). However, such a generic active learning approach typically needs many interactions and easily becomes cumbersome for the user [2].

For this challenge of selecting and reusing ML solutions,

the literature also describes several tools that implement semi-supervised ML with interactive user input. In these tools, the biologists interactively provide annotations for part of the data set on which a model is trained. The trained model is then applied to the rest of the data set. The Waikato Environment for Knowledge Analysis (WEKA) [3] is an interactive, semi-supervised machine learning tool for image pixel classification. Some example objects are annotated through a graphical user interface to support training. The user can interactively provide feedback by correcting or adding labels or annotations until the trained model exhibits satisfying performance and can be applied to the overall data set. Ilastik is another semi-supervised tool that leverages machine learning to segment, classify, track and count cells [4]. Also Ilastik is based on annotation of only a part of the data set. Compared to other ML segmentation tools, both WEKA and Ilastik reduce the size of the required training data set. Nevertheless, these semi-supervised tools still require the user to annotate objects that are informative for training based on visual impression.

We propose a novel combination of auto-tuning and active learning to ease the selection of ML solutions as well as their parameter tuning effort for scientists. In our approach, we cluster the input data and tune the parameters separately for each cluster. Furthermore, we leverage that in some use cases, although no objective function for auto-tuning is available that covers all aspects of result quality, we can identify a metric that approximates the quality of prediction results. In the case of biological image segmentation, while there is no metric to assess the overall quality of segmentation results, we can define a metric that can preliminarily assess the quality of results by comparing the segmented object with the pixels in its neighbourhood. A core idea of this paper is that such a metric helps us with the choice of candidates to show to the user in active learning and thus reduces the number of interactions with the user.

We especially focus on a use case from biology, namely the object segmentation in biological images. Here, we provide a framework which first maps the objects into different clusters and provides selection and adjustment of the segmentation solutions per cluster.

The contributions of our paper are summarized as follows

- We propose a framework for semi-supervised ML with interactive user input that enables scientists to select ML solutions and tune their parameters, with few interactions, in scenarios where no labelled training data and no objective function are available, but where a metric that approximates quality can be defined.
- We provide such a metric for the use case of segmentation in biomedical images.
- We conduct a biological case study in which users can obtain satisfying results with effectiveness and efficiency in a few interactions.

The expected benefits of our approach is that scientists who are not ML experts can more easily reuse and deploy ML solutions with high result quality. While we have evaluated our approach for the use case of image segmentation only so

far, we expect that it is applicable for reusing and tuning of ML solutions where (1) no labelled training data and no objective function are available, (2) a global parameter settings would not work for all the objects and the parameters ideally need to be tuned separately for different input object types and (3) a metric that approximates quality can be defined.

The rest of the paper is structured as follows. In Section II, we present our specific use case, segmentation in biomedical imaging. A brief summary of the segmentation metrics is provided in Section III, followed by a demonstration of our proposed framework in the next section (Section IV). We introduce research questions and some information regarding the evaluation and experiment on Section V. We then discuss our results in the Section V-A and threats to validity in the next section. The related works are studied in Section VII and our conclusions are given in Section VIII.

II. SCIENTIFIC USE CASE

In biomedical images, the analysis of objects usually involves quantifying multiple properties, for instance, object shape, object location, or the intensity of an experimentally applied label. In many cases, these measurements are based on the segmentation of objects, meaning the assignment of image pixels (or voxels) as “object” (1) or “not-object” (0). Often, biological samples are treated with a target-specific labeling substance, which provides high intensity levels that can be exploited in ML-based object segmentation. A range of commonly used segmentation algorithms exists, which determine an intensity threshold above which pixels are considered “object”. These segmentation algorithms, however, typically require adjustment to specific samples and labeling techniques. A well-chosen and well-adjusted segmentation algorithm is prerequisite for robust and precise segmentation, forming the basis for analysis steps.

One of the simplest unsupervised intensity-based AI segmentation approaches is thresholding [5]. Such algorithms calculate a threshold intensity based on the intensity distribution in the image [6] and the setting of the algorithm can be only adjusted manually by modifying the script. The threshold intensity then is applied to separate pixels into two categories, pixels with greater intensity than the threshold (assigned 1) and pixels with lower intensity values (assigned 0). Other segmentation methods are either in the form of interactive pipelines with user-defined parameters (through user interactions) [4] or deep networks (supervised) with automated parameter tuning on the basis of large data sets [7].

For biologists, choosing between many supervised and unsupervised ML segmentation solutions and deployment the algorithm in their pipeline is challenging. For example, if the microscopy images are of good quality and objects with clear boundaries are to be analyzed, conventional unsupervised thresholding solutions can be a good choice. However, if the intensity profiles are obscured by noise or objects without sharp boundaries and variable in size, intensity and morphology (Fig. 1) [8] need to be analyzed, conventional AI methods usually fail to properly segment the objects of interest [6]. In

contrast, supervised ML (fully-automated) segmentation algorithms which automatically adapt the parameter based on the given input and the annotated output are more robust against the variety in the objects' size, intensity and morphology. However, these supervised methods require training data sets consisting of annotated images. These annotated data sets must be sufficiently comprehensive to avoid over-fitting, which compromises segmentation of unknown images. Providing such manually created training annotations is labour-intensive, and requires knowledge of domain experts. The same limitation arises when a supervised method needs to be re-trained for new data sets, which typically also requires additional annotation by expert biologists [9]. There are also semi-supervised ML segmentation algorithm with interactive user input. In these tools, users interactively provide annotations for part of the data set and the software later segments the unseen data [3], [4]. However, these tools still require the user to annotate objects that are informative for training.

Our idea is to ease the selection and the deployment of ML segmentation solutions by a framework that firstly searches through the existing unsupervised segmentation solutions and secondly optimises the parameter setting with the help of user input seems to be a more practical approach in the bioimaging pipeline where manually annotating is in practice infeasible. However, it is important to make the interaction as short as possible, to encourage end-users to deploy the algorithm in their imaging pipeline. Here, we propose a framework integrating autotuning of the unsupervised segmentation with the user interactions, which enables biologists to easily find, deploy and adjust the existing segmentation methods on their own data. Underlying this framework is a mapping from object-morphology feature space to the specific segmentation algorithm and parameters applied during the segmentation process. This mapping is iteratively adjusted by an alternation between unsupervised (without additional user interactions) and semi-supervised (with additional user interactions) training episodes, which is optimized to reduce the number of user interactions. In other words, we ask the user to select from alternative example segmentation, thereby adjusting the segmentation algorithm per object type, with a few user requests as possible. We apply our framework to the example case of RNA Polymerase II (Pol II) segmentation in 2D super-resolution microscopy images. These clusters exhibit highly variable morphologies with diffuse boundaries, providing a well-suited example case.

III. BACKGROUND: SEGMENTATION METRICS

Metrics to evaluate image segmentation can be categorized based on different criteria. One criterion is whether a metric is considered as subjective or objective [11]. Subjective metrics are based on human visual assessments. These metrics are, by definition, not formally defined. To reach a generalizable result, the number of observers should be high.

Another criterion is whether the evaluation of a segmentation approach is analytical or empirical [12]. In empirical evaluations, the accuracy and precision are evaluated on the

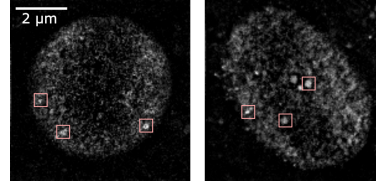


Fig. 1. **Shape, intensity and contrast of foci varies from one to another** Images are the nuclear mid-plane of a fixed zebrafish embryo at sphere stage, which are recorded by STEDD super-resolution microscopy. Intensity distributions of RNA Polymerase II (Pol II) Serine 5 phosphorylation (Ser5P) were obtained by STEDD microscopy, while Pol II Ser2P intensity distributions are obtained by regular confocal microscopy from the same plane. Pol II Ser5P clusters are marked. Intensity scale from black to white adjusted to the 0.01-th and the 99.99-th percentile [10]

basis of example data in a supervised fashion. An analytical evaluation does not rely on example data, but rather analyzes the theoretical properties of the segmentation [13].

As a third criterion, the evaluation of image segmentation can be supervised (with annotated images) or unsupervised (without annotated image). Supervised evaluation benefits from the direct comparison between the segmentation results and the reference annotated image. The evaluation can be based on per-pixel comparison (True Positive (TP) and True Negative (TN), Matthews Correlation Coefficient (MCC) and F-measure, region-based comparison (Bipartite Graph Matching (BGM) [14] and Segmentation Covering (SC) [15]) or distance-based evaluation (Hausdorff Distance, directional Hamming distance and Mahalanobis distance [16]). Unsupervised metrics evaluate the quality of segmentation by directly calculating the feature parameters of the segmentation result without using the annotated reference image. These metrics are the ideal kinds of metric especially when there is no ground truth. These unsupervised metrics are mostly based on mathematical indicators showing the quality of segmentation results, such as Peak Signal to Noise Ratio [17], calculating the inside and outside contrast of the segmentation area, or measuring the foreground and background variances [17].

IV. METHOD

In this section, we propose our framework (Algorithm 1, Fig.2) for the semi-automated deployment of image segmentation solutions. The first step is object set preparation, where the object positions are spotted and a bounding box around the region-of-interest (ROI) is extracted and clustered into some main categories. Within each object category, the choice of segmentation solutions and the parameter adjustments are performed. In other words, the framework initially starts with 9 segmentation solutions in each object category, and an unsupervised metric (Section. IV-D) is later used to preliminary filter segmentation solutions with bad results and picks some of the better-performed solutions and the related results are presented to the users (Section. IV-C) for final decision. The framework then proceeds to iteratively tune the (hyper) parameters of the selected segmentation solution via the presentation of selected segmented objects (Section. IV-C).

Algorithm 1 The proposed semi-automated framework for bio-imaging segmentation

INPUT:

Streaming images $\{x_i\}_{i=1}^n$

Base unsupervised segmentation algorithms

$\vec{\Phi} = \{\phi_j(x_i, \vec{\lambda}_j)\}_{j=1}^m$

OUTPUT:

Optimum values for $\vec{\lambda}_j$ which is the parameters and hyperparameters associated with segmentation algorithm $\phi_j(\cdot)$

- 1: Prepare a dataset of objects according to Section IV-A
- 2: Cluster objects in different categories based on object morphologies (Section IV-B)
- 3: Initiate $\{\vec{\lambda}_j\}_{j=1}^m$
- 4: **while**
- repeat for each morphology class until convergence **do**
- 5: Apply $\vec{\Phi} = \{\phi_j\}_{j=1}^m$ to i th object (o_i),
 $s_{ij} = \phi_j(o_i, \vec{\lambda}_j)_{i=1}^m$
- 6: Filter segmentation results ($\{s_{ij}\}_{j=1}^m$) using metric described in Section IV-D
- 7: Select the k best segmentation results (s_{ij}) for each object
- 8: Show the selected results to the user
- 9: Optimise the pipeline based on the user selection
- 10: Update set $\vec{\Phi}$ and $\vec{\lambda}$
- 11: **end while**
- return** $\vec{\Phi}, \vec{\lambda}_j, \vec{s}$

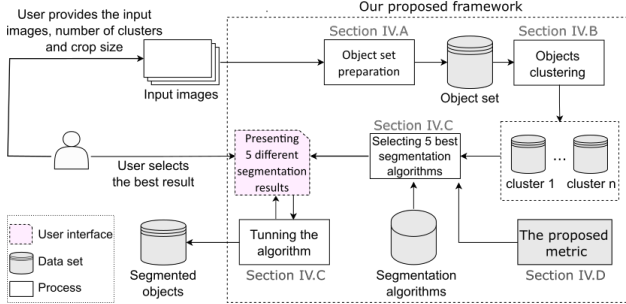


Fig. 2. The proposed framework consists of several automated steps. At the beginning, the user uploads the input raw images and provides the number of object clusters and the crop size. With a few interactions, the segmented objects are given back to the user.

A. Object set preparation

In this step, the framework spots and crops the bounding box around segmentation-target objects. Some simple thresholding algorithms can successfully detect the position of the target objects, however, depending on the experiments there might be some other bright subcellular components making directly spotting the region of interest (ROI) difficult (Fig. 1). The framework therefore at first employs a filter to make the image blurry and then proceeds with a thresholding technique to only spot the ROIs. Generally, any blurry filter can be used, however, in our particular application case, a Gaussian blur

filter with a standard deviation of 6 pixels is applied on each nucleus, followed by a local thresholding method to roughly spot the ROIs. A region of size (60×60) is then cropped and saved (Fig. 3).

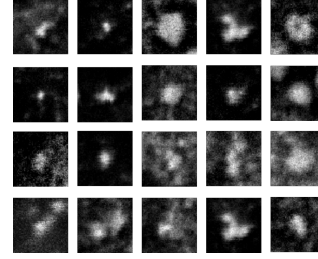


Fig. 3. Detail views of objects. Representative Pol II Ser5P clusters mid-section obtained by STEDD super-resolution microscopy from a fixed sphere-stage zebrafish embryo. Each image is of size 60×60 pixels, The images are randomly selected from a dataset of size 148

B. Feature space representation of object set

In machine learning, every measurable (mostly numeric) property describing a characteristic of an object is generally called feature. Each segmented-target object can be represented by a vector of features (min intensity, max intensity, mean intensity, standard deviation, raw pixel values, ...) and feature space is the space spanned by the feature vectors. To enable the choice of different algorithms for different types of objects, the framework first establishes a classification of the overall object sets into subgroups. Generally, any method of classification can be used. In our particular application case, previous work has proposed a classification into 3 morphological types [10]. Type i objects appear similar to a dot, type ii objects are larger and relatively compact, type iii objects are larger and unfolded in shape. Type i and type ii objects have clear boundary lines, while type iii objects do not offer a clear boundary and exhibit extensive morphological variety (Fig. 3). Therefore, the segmentation algorithm and parameters can be expected to differ between type iii and other object types.

In our particular application case, the framework uses K-nearest neighbor (kNN) algorithm to group objects in 2 subgroups (one subgroup with type iii objects and another subgroup contains remaining objects).

C. Semi-automated choice and adjustment of segmentation algorithm

For each cluster, several segmentation algorithms with default settings (Table I) are employed to segment objects categorised in that specific cluster. In principle, any segmentation algorithm can be used here. For simplicity, in our application case, the framework only considers segmentation methods based on the thresholding technique.

The framework then uses an unsupervised metric (Section IV-D) to select the 5 best segmentation algorithms. Then it randomly selects a few objects and presents the selected

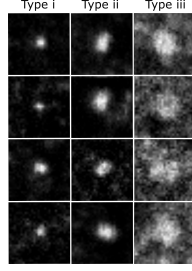


Fig. 4. Detail views of the object of the 3 apparent morphology Type i objects are small and appear similar to a dot, type ii objects are larger and compact, type iii objects are also larger, appear unfolded, and have blurred boundaries.

objects with the segmented results obtained by the 5 best-performed segmentation algorithms to the user. The user then selects the best algorithm by looking at their performance on a few objects (Fig.5).

TABLE I
VARIOUS THRESHOLDING METHODS. THRESHOLDING TECHNIQUES AIM TO CREATE A BINARY MASK FROM A GRAYSCALE IMAGE, WHICH SEGMENTS OBJECTS FROM A BACKGROUND.

Algorithm	Short description
Isodata	A threshold is iteratively found based on the image histogram
Mean	Uses the mean value of pixel intensities as threshold value
Minimum	Takes a histogram of the image and smooths it repeatedly until there are only two local maxima [18].
Otsu	Based on maximizing the variance between two classes of pixels which are separated by the threshold [19].
Triangle	A geometric method assuming a maxima near one end of the histogram and searches towards the other end [20].
Yen	Based on Yen's thresholding method [21]
Li	An iterative method based on Li's Minimum Cross Entropy thresholding method [22]
Local	Each threshold value for each pixel is the weighted mean of the local neighborhood minus an offset value.
Median	Uses the median value of pixel intensities as threshold value

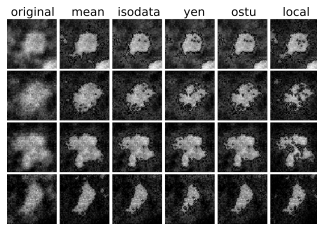


Fig. 5. Results obtained by 5 different segmentation algorithm. Four objects (each is represented in one row) are randomly selected from cluster iii of size 117. A set of 9 segmentation algorithms are initially applied on the object set and 5 best segmentation algorithms (each in one column) are selected and the results are presented to the user for the final selection.

For the selected algorithm, the framework repeatedly chooses 5 different parameter sets, and presents the resulting segmentation masks for selection by the user (Fig. 6). Using deepening search space strategy the framework every time narrows down to the search space around the selected setting

and provides 5 different choices for the user. This procedure is repeated until receiving input from the user that satisfactory segmentation results have been obtained. The framework performs this iterative adjustment for every cluster separately, so that the optimum settings can differ per cluster.

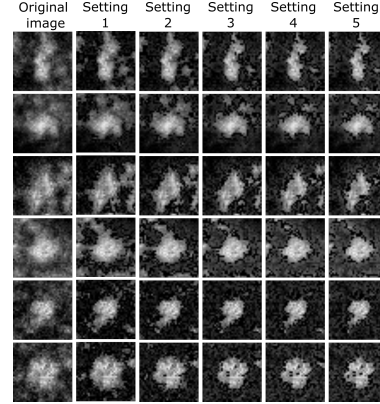


Fig. 6. Results obtained by the selected segmentation algorithm with 5 different parameter settings. Six objects (each is represented in one row) are randomly selected from cluster iii of size 117. The selected segmentation algorithm with 5 different settings is applied to the object sets and the results are presented to the user from the loose to tight segmentation. User chooses the best setting and every time the search space is narrow down to the region around the selected setting.

D. Metric

To figure out how good the segmentation is for an object (O_i), the framework starts from a very tight segmented mask ($Mask_i$) and every time dilates the segmented mask ($Mask_{i+1}$) and calculates the average intensity of the difference area between the original mask and the dilated one $d_i = \text{mean}\{(Mask_{i+1} - Mask_i) \times O_i\}$ (Fig. 7). This dilation is repeated 5 more times and at each time the framework calculates the average intensity of the difference area (d_i). In general, the average intensity $\{d_i\}_{i=1}^5$ is every time decreased and the segmentation line should be close to wherever the decrease rate is maximised.

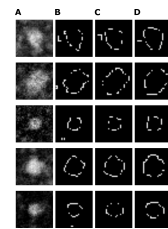


Fig. 7. Segmentation line should be close to wherever that the intensity drops A: representative Pol II Ser5P clusters mid-section obtained by STEDD super-resolution microscopy from a fixed sphere-stage zebrafish embryo. Each image is of size 60×60 pixels. B, C, D: dilated morphologies. from left to right, each time the morphology is dilated more.

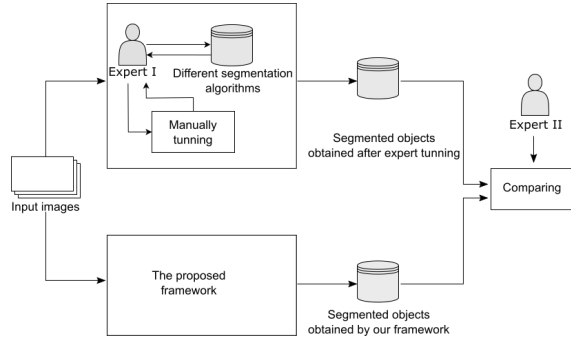


Fig. 8. A subjective metric is used for evaluation. The results obtained by our framework is compared by the results obtained after tuning the algorithm by an expert. This evaluation is based on a human visual assessment.

V. EVALUATION

In this section, we present a biological case study, which allowed us to evaluate our framework based on the following two research questions:

- RQ1: How effective is our framework in improving the quality of segmentation?
- RQ2: How much interactions the framework needs from users to deploy the ML solutions?

We conduct all procedures on a laptop with Intel(R) Core(TM) i7-8665U CPU 1.90GHz 2.11 GHz processor and 32.0 GB RAM, using the Microsoft Windows operating system. The code is written in Python, uses the “scikit-learn” package, and is publicly released on here.

A. Experimental setup

We used input images recorded by STEDD microscopy in a previous study [10], [23]. The images are maximum-intensity projections of cell nuclei in fixed zebrafish embryos, where Pol II Serine 5 phosphorylation was labeled by indirect immunofluorescence (Fig. 1) The data set contained images of 60 nuclei. We extract an object set based on cropped region-of-interest images of 60×60 pixel size, which are centered on prominent groups of Pol II. We obtained a set of 148 objects that were used for the further application of our framework.

To address the RQ1 we ask three biologists to use our script for the adjustment of object segmentation. In line with previous work, objects are automatically pre-categorised in 2 clusters based on image features that do not rely on segmentation (Fig. 4) [10]. For each cluster, 9 segmentation algorithms (Table. I) are applied to the objects. Five of the best-performing segmentation algorithms are selected using the metric introduced in section IV-D. These selected algorithm are then applied on a handful randomly-selected objects and the results are then shown to the users (Fig. 5). The best-performing algorithm is then selected by the user based on the visual presentation of the few selected objects.

Afterwards, the framework proceeds with tuning the parameters of the selected algorithm by deepening space search. Optimal parameter values are then obtained by further user

interactions (Fig. 6). Once the framework has satisfyingly tuned the segmentation parameters within the current cluster of objects, the same procedure is executed for the next cluster, until all clusters are adjusted.

To establish a reference for the evaluation of our training results, we asked one expert to choose a segmentation algorithm based on the object set globally (considering all the clusters together) and manually tune the parameter of the algorithm. The expert had a wide knowledge of segmentation algorithms and was very familiar with the application data and the adjustment for this expert took around 2 hours. The results that we obtained from the expert, can be considered as the best results that can be achieved by the use of the existing resource. We then present 10 randomly selected pairs of segmented objects obtained by our script and after manual tuning by the expert to another expert to evaluate the quality of segmentation¹. Using an expert human to visually assess the result is a kind of subjective metric. We provided four choices for each comparison pair: (i) segmented image A is better, (ii) segmented image B is better, (iii) both segmentation results are almost equal and (iv) the evaluation of segmentation depend on an additional consideration formulated by the expert. We randomly alternated the assignment to choices A and B to prevent bias in choice. The evaluation indicates that, in the eyes of an expert, in 90% of cases the segmentations obtained by our framework are better than or almost equal to the segmentations obtained after manual tuning by an expert (Table II).

TABLE II
EVALUATION RESULTS PERFORMED BY THE EXPERTS

Choice	Percentage
Results obtained by our framework outperform	80%
Results obtained after expert tuning outperform	10%
Both results look almost similar	10%
It depends on ...	0%

To address the RQ2, we ask three biologists to use our script and we record the time spent and the number of interactions required for tuning the script. On average, the script converges within 8 user interaction within 6 minutes (Table III). The simplicity of our proposed framework was rated high (6 out of 7, Table III).

VI. THREATS TO VALIDITY

The validity of our conclusions could be mainly compromised by the example application and evaluation setting. We use only example data from one type of biological sample, recorded on one type of microscope. The experiment contains different samples to account for day-to-day variability in sample quality. However, image characteristics, e.g., intensity distribution and image resolution, can differ fundamentally between different microscopes. Adding data from other types of samples and microscopes would strengthen the validity of our findings.

¹A sample questionnaire presented to the expert is available at [link](#).

TABLE III
NUMBER OF INTERACTIONS AND THE DEDICATED TIME REQUIRED FOR CONVERGENCE

	Total number of interactions	dedicated time	Simplicity 1: not simple 7: maximally simple	Segmentation algorithm selected for cluster iii	Segmentation algorithm selected for other objects
Case study 1	8	6 minutes	6	Isodata	Ostu
Case study 2	10	7 minutes	6	Isodata	Isodata
Case study 3	6	5 minutes	7	Local	Ostu
Average	8	6 minutes	6		

Three expert biologists working in the same laboratory tested the framework by tuning segmentation on the object set. Bias could be further reduced by increasing the number of users, and also recruit users from different research groups.

The third threat to our experiment design is due to the number of experts assessing the segmentation quality. We used only one expert to visually assess the quality of segmentation results. Assessment by a larger panel of experts would ensure a more reliable and unbiased result.

VII. RELATED WORK

Our work is mainly related to the interactive approaches with user-guided parameter selection. It is almost a decade since interaction object segmentation has firstly emerged. The early approaches are mostly formulated as energy minimization tasks on a graph. An image is first represented as an undirected graph, which can be defined by a joint probability p . The goal is to assign a label to each pixel (0 represents the background and 1 for the objects) in a way that the probability p is minimized [24]–[27].

There have been some other interactive segmentation works that require users to annotate the edges (extreme-left, extreme-right, extreme-top and extreme-bottom) of objects. The connection lines between edges represents boundary line. Training on the annotated objects, the method can hopefully later detect the boundaries of unseen objects [28], [29]. The region inside the boundaries are considered as foreground (object) and the outside area is labelled as background. There are several major challenges regarding the transfer of these methods in the biological subcellular segmentation tasks. Firstly, some subcellular components cannot be inherently represented by edges and secondly, the whole region inside the boundaries sometimes is not the object, for example, objects with the shape of doughnuts.

Recently, deep networks have been extensively used in the interactive segmentation tasks [30], [31]. Some deep learning tasks are on the basis of detecting the four edges, where for a part of the dataset, users annotate the edges and the network is later trained to detect the edges for unseen input [32]. There are some other interactive deep network methods for image segmentation [33], [34], where users should provide bounding boxes for a part of the data set and a network is learned to detect the bounding box for the rest of the dataset. These approaches are sometimes combined with reinforcement learning which asks users to mark the foreground. These approaches have been successfully used in CT/MRI image segmentation,

where the organ appears in similar shapes in all images. In some studies [35], [36], the user inputs (bounding box annotations) have been combined with the graph convolutional networks (GCN). In the training phase, the manually-annotated data is used and the GCN learns to represent an object with a polygon around it. To refine the trained model, the predicted polygon can be further adjusted by the user interaction.

However, there is a general limitation in the use of methods on the basis of deep networks, which is the requirement of annotated images. To avoid over-fitting, the annotated data sets must be sufficiently comprehensive, which makes it even more challenging. Providing such manually created training annotations is time-consuming and requires the knowledge of domain experts. The same limitation arises when a network needs to be re-trained for new data sets, which typically also requires additional annotation by experts [37]. There is also a particular challenge in the transfer of these approaches to biological image segmentation where the number of objects that exists in one image can be very high. For example, sometimes a single image may contain 200 cells and it is practically infeasible to annotate the bounding boxes or the edges. The next limitation in regard to biological image segmentation is the high degree of variety in object shapes, when the object of interest appears in very complex object shapes with unclear edges and boundaries (which is inherently very common in biological subcellular components) [38]. Having reviewed the literature, we realized that a practical approach for biological segmentation is a method keeping the user interaction short and simple while enabling adjustment for unseen data.

VIII. CONCLUSION

Detecting the different parts that build a biological cell is one of the core tasks in the analysis of biomedical images. There are some automated approaches that can automatically detect the objects, however, the deployment of these algorithms for the processing of unknown data sets poses a major challenge in practical application. The main challenge is the need for annotated objects, which can be provided by experienced domain specialists, which is very labour expensive. In this paper, we proposed a framework that enables object-type specific adjustment of the segmentation algorithm within the face of unseen data with a few user interactions. Underlying this framework was a mapping function between object-morphology feature space and specific algorithm settings. We then applied this framework to a biological case study and

validated the performance and simplicity of our proposed framework through the case study.

In the future work, we plan to explore how object morphology feature space may affect the next step of our framework and accordingly improve the feature space representation of object set by adopting an interactive fashion instead of an unsupervised manner. We also plan to assign an uncertainty value to each segmented object and present the objects with high certainty for user selection.

ACKNOWLEDGEMENT

This work is supported by the Helmholtz Association under the joint research school “HIDSS4Health – Helmholtz Information and Data Science School for Health” and the Helmholtz program Natural, Artificial, and Cognitive Information Processing (NACIP). HH was supported by HIDSS4Health. We thank Irina Mamontova, Roshan Prizak and Agnieszka Pancholi for help in conducting the case study. We also thank Christopher Gerking for comments on our manuscript.

REFERENCES

- [1] J. Segal and C. Morris, “Developing scientific software,” *IEEE Software*, vol. 25, no. 4, pp. 18–20, 2008.
- [2] S. E. Walker, “Active learning strategies to promote critical thinking,” *Journal of athletic training*, vol. 38, no. 3, p. 263, 2003.
- [3] I. Arganda-Carreras, V. Kaynig, C. Rueden *et al.*, “Trainable weka segmentation: a machine learning tool for microscopy pixel classification,” *Bioinformatics*, vol. 33, no. 15, pp. 2424–2426, 2017.
- [4] S. Berg, D. Kutra, T. Kroeger *et al.*, “Ilastik: interactive machine learning for (bio) image analysis,” *Nature Methods*, vol. 16, no. 12, pp. 1226–1232, 2019.
- [5] E. Dougherty, *Mathematical morphology in image processing*. CRC press, 2018, vol. 1.
- [6] S. Cantaloube, K. Romeo, P. Le Baccon *et al.*, “Characterization of chromatin domains by 3d fluorescence microscopy: an automated methodology for quantitative analysis and nuclei screening,” *BioEssays*, vol. 34, no. 6, pp. 509–517, 2012.
- [7] J. Yi, H. Tang, P. Wu, *et al.*, “Object-guided instance segmentation for biological images,” in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 34, no. 07, 2020, pp. 12677–12684.
- [8] S. Osterwald, K. I. Deeg, I. Chung, D. Parisotto, and othersn, “Pml induces compaction, trf2 depletion and dna damage signaling at telomeres and promotes their alternative lengthening,” *Journal of cell science*, vol. 128, no. 10, pp. 1887–1900, 2015.
- [9] J. C. Caicedo, A. Goodman, K. W. Karhohs, B. A. Cimini, J. Ackerman, M. Haghighi, C. Heng, T. Becker, M. Doan, C. McQuin *et al.*, “Nucleus segmentation across imaging experiments: the 2018 data science bowl,” *Nature methods*, vol. 16, no. 12, pp. 1247–1253, 2019.
- [10] A. Pancholi, T. Klingberg, W. Zhang, R. Prizak, I. Mamontova, A. Noa, M. Sobucki, A. Y. Kobitski, G. U. Nienhaus, V. Ziburdaev *et al.*, “Rna polymerase ii clusters form in line with surface condensation on regulatory chromatin,” *Molecular systems biology*, vol. 17, no. 9, p. e10272, 2021.
- [11] Y. Chen, D. Ming, L. Zhao *et al.*, “Review on high spatial resolution remote sensing image segmentation evaluation,” *Photogrammetric Engineering & Remote Sensing*, vol. 84, no. 10, pp. 629–646, 2018.
- [12] Y. J. Zhang, “A survey on evaluation methods for image segmentation,” *Pattern recognition*, vol. 29, no. 8, pp. 1335–1346, 1996.
- [13] Z. Wang, E. Wang, and Y. Zhu, “Image segmentation evaluation: a survey of methods,” *Artificial Intelligence Review*, vol. 53, no. 8, pp. 5637–5674, 2020.
- [14] A. A. Taha and A. Hanbury, “Metrics for evaluating 3d medical image segmentation: analysis, selection, and tool,” *BMC medical imaging*, vol. 15, no. 1, pp. 1–28, 2015.
- [15] N. Dey, V. Rajinikanth, A. S. Ashour, and J. M. R. Tavares, “Social group optimization supported segmentation and evaluation of skin melanoma images,” *Symmetry*, vol. 10, no. 2, p. 51, 2018.
- [16] J. Pont-Tuset and F. Marques, “Supervised evaluation of image segmentation and object proposal techniques,” *IEEE transactions on pattern analysis and machine intelligence*, vol. 38, no. 7, pp. 1465–1478, 2015.
- [17] H. Zhang, J. E. Fritts, and S. A. Goldman, “Image segmentation evaluation: A survey of unsupervised methods,” *computer vision and image understanding*, vol. 110, no. 2, pp. 260–280, 2008.
- [18] J. M. Prewitt and M. L. Mendelsohn, “The analysis of cell images,” *Annals of the New York Academy of Sciences*, vol. 128, no. 3, pp. 1035–1053, 1966.
- [19] N. Otsu, “A threshold selection method from gray-level histograms,” *IEEE transactions on systems, man, and cybernetics*, vol. 9, no. 1, pp. 62–66, 1979.
- [20] G. W. Zack, W. E. Rogers, and S. A. Latt, “Automatic measurement of sister chromatid exchange frequency,” *Journal of Histochemistry & Cytochemistry*, vol. 25, no. 7, pp. 741–753, 1977.
- [21] M. Sezgin and B. Sankur, “Survey over image thresholding techniques and quantitative performance evaluation,” *Journal of Electronic imaging*, vol. 13, no. 1, pp. 146–165, 2004.
- [22] C. H. Li and C. Lee, “Minimum cross entropy thresholding,” *Pattern recognition*, vol. 26, no. 4, pp. 617–625, 1993.
- [23] L. Hilbert, “Analysis of RNA polymerase II phosphorylation in STimulated Emission Double Depletion (STEDD) microscopy images,” Jun. 2021. [Online]. Available: <https://doi.org/10.5281/zenodo.4973062>
- [24] X. Bai and G. Sapiro, “Geodesic matting: A framework for fast interactive image and video segmentation and matting,” *International journal of computer vision*, vol. 82, no. 2, pp. 113–132, 2009.
- [25] Y. Y. Boykov and M.-P. Jolly, “Interactive graph cuts for optimal boundary & region segmentation of objects in nd images,” in *Proceedings eighth IEEE international conference on computer vision. ICCV 2001*, vol. 1. IEEE, 2001, pp. 105–112.
- [26] V. Gulshan, C. Rother, A. Criminisi, A. Blake, and A. Zisserman, “Geodesic star convexity for interactive image segmentation,” in *2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*. IEEE, 2010, pp. 3129–3136.
- [27] S. Cagnoni, A. B. Dobrzeniecki, R. Poli, and J. C. Yanch, “Genetic algorithm-based interactive segmentation of 3d medical images,” *Image and Vision Computing*, vol. 17, no. 12, pp. 881–895, 1999.
- [28] V. Kwatra, A. Schödl, I. Essa, G. Turk, and A. Bobick, “Graphcut textures: Image and video synthesis using graph cuts,” *Acm transactions on graphics (tog)*, vol. 22, no. 3, pp. 277–286, 2003.
- [29] D. P. Papadopoulos, J. R. Uijlings, F. Keller, and V. Ferrari, “Extreme clicking for efficient object annotation,” in *Proceedings of the IEEE international conference on computer vision*, 2017, pp. 4930–4939.
- [30] N. Xu, B. Price, S. Cohen, J. Yang, and T. S. Huang, “Deep interactive object selection,” in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2016, pp. 373–381.
- [31] Z. Wang, D. Acuna, H. Ling, A. Kar, and S. Fidler, “Object instance annotation with deep extreme level set evolution,” in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2019, pp. 7500–7508.
- [32] K.-K. Maninis, S. Caelles, J. Pont-Tuset, and L. Van Gool, “Deep extreme cut: From extreme points to object segmentation,” in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 616–625.
- [33] L. Castrejon, K. Kundu, R. Urtasun, and S. Fidler, “Annotating object instances with a polygon-rnn,” in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 5230–5238.
- [34] H. Ling, J. Gao, A. Kar, W. Chen, and S. Fidler, “Fast interactive object annotation with curve-gcn,” in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2019, pp. 5257–5266.
- [35] V. Caselles, R. Kimmel, and G. Sapiro, “Geodesic active contours,” *International journal of computer vision*, vol. 22, no. 1, pp. 61–79, 1997.
- [36] D. Acuna, A. Kar, and S. Fidler, “Devil is in the edges: Learning semantic boundaries from noisy annotations,” in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2019, pp. 11075–11083.
- [37] M. H. Hesamian, W. Jia, X. He, and P. Kennedy, “Deep learning techniques for medical image segmentation: achievements and challenges,” *Journal of digital imaging*, vol. 32, no. 4, pp. 582–596, 2019.
- [38] M. Jahanifar, N. A. Koohbanani, and N. Rajpoot, “Nuclick: From clicks in the nuclei to nuclear boundaries,” *arXiv preprint arXiv:1909.03253*, 2019.