Active contour and deep learning methods for single-cell segmentation in microscopy images

PhD Thesis

Ervin Tasnádi
Supervisor: Peter Horvath, PhD

Doctoral School of Computer Science

Faculty of Science and Informatics
University of Szeged

Szeged, 2023

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Microscopy image segmentation

Many of the experiments in life sciences can be analyzed using imaging technologies, thus microscopy has become one of the fundamental tools in the pipelines of most experiments in this field. This kind of analysis is often used to solve a specific task such as counting the number of cells in images taken from cell cultures or biopsies from human tissues, and quantify their shapes or textures that assumed to have important biological or medical relevance. Manually analyzing the images generated from experiments using microscopy technology was not a big challenge until the recent years: it was part of the daily life of a researcher. The manual approach has two main drawbacks: experts have to spend their valuable time to perform these tasks manually (for example counting instances), but other tasks, for example, classification of the instances based on their morphology has some subjective bias among different scientist. With the emergence of systems biology and the automatization of imaging pipelines, thousands of samples can be acquired daily with high resolution microscopes. Thus, analyzing this amount of data is no longer practically feasible without efficient algorithms that aim to also automatize the analysis tasks. Since most of the measurement pipelines start from the quantification of nuclei instances e.g. segmentation of the nuclei [13], we aim to concentrate on nuclei segmentation of biological images. That is, we are interested in separating the pixels of a microscopy image into two classes: the pixels that are part of the nuclei instances and the pixels that are not. Furthermore we also want to group the pixels of the instances of the same group if they are representing the same object. We call the first task as semantic segmentation and the latter is the instance segmentation. Several software were developed for biological image analysis that includes the segmentation task. Such examples are the CellProfiler [15, 50] and ImageJ[2, 93]/Fiji[91]. These software offer classical image processing algorithms such as thresholding and watershed segmentation [8, 9]. The user can define pipelines and adjust the param-

eters for a particular experiment. More advanced methods such as active contours are also used for segmenting nuclei or cell instances. [5]. Active contour methods offer a clear advantage over basic image processing primitives such as thresholding or region growing since the contour is regularized by ensuring its smoothness property thus the noise or non-relevant object parts can be excluded easily. These contour based approaches were extended with shape priors for segmenting microscopy images such as: circular shapes [67, 68], overlapping objects [66], spherical and ellipsoid-like objects [70], or even higher-order priors that cut object clusters into single objects. [69]. More recent methods use supervised approaches for microscopy image segmentation. One of such examples is Ilastik [7, 97], that uses a random forest[11] classifier to classify pixels based on their features while using minimal pixel annotation. Later, multi-layered fully convolutional neural networks [58] were applied for pixel classification of microscopy images. [88] These networks do not use a two phase (feature extraction then classification) approach but they learn the feature extractor from the data directly by adjusting the weights of the convolutions in each layer: the two stages are unified into a fully differentiable function ("end-to-end learning" concept). Furthermore, thanks to the stacked convolution operations, the network can condition the class of a single pixel on the features of pixels coming from a much larger neighborhood constructing object clues conditioned on more global information. The current snapshot of the field is summarized in a recent review paper. [38] Nuclei segmentation is not only interesting in 2D images but also in 3 dimensions. Biological samples extracted from species are inherently 3-dimensional and there is growing interest in creating *in-vitro* cell populations in 3D as they might be a better model of human tissues compared to 2D cell populations. [24] Recent microscopy technology allows us to precisely image samples at multiple z-levels with high axial resolution. The combination of such planar images leads to volumes that can be analyzed as 3D volumes. A recent review summarizes current methods for nuclei segmentation of 3D samples. [23, 82]

This thesis introduces methods developed for the above problems.

1.2 Structure of the document

The results are split into two parts:

- part I deals with active contour methods and level sets and introduced in chapter 2, the results are in chapters 3, 4, 5 while

- part II summarizes our results in microscopy image segmentation with deep learning introduced in chapter 6 and the results are in chapters 7, 8.

1.3 Thesis points

This section connects the thesis points and the chapters. Each chapter is supported by at least one publication. See the summary for an overview of the connection between the supporting publications and chapters.

- **In our first thesis point**, we propose a fast and accurate level set reinitialization method using the phase field functional that forms a local smooth phase transition in the vicinity of the contour. The issue with the phase field however, that it has a curvature driven component that moves the contour as a side effect. We eliminate this side effect from the phase field functional by combining the gradient term with a second order smoothness. We empirically show that the side effect is minimal compared to the competing methods while computationally inexpensive. The results are presented in chapter 3.

- **In the second thesis point** we introduce a 3D selective active contour model developed for the analysis of 3D fluorescent images. The model uses prior object parameters to selectively retrieve objects. We show how the priors work on synthetic and real microscopy data. We also propose an open source annotation software called *3D-Cell-Annotator*, that uses the developed method and the reinitialization method proposed in 3 combined with a local region data term in 3D with efficient local level set implementation. We show that using the proposed software, accurate segmentation can be created with significantly less user effort. The selective model is presented in chapters 4 while the annotation software is presented in chapter 5.

- **The third thesis point** introduces two automatic augmentation protocols for nuclei segmentation of 2D images (multiple microscopy modalities) with convolutional neural networks. The first model, called *nucleAIzer* simulates microscopy masks by cell population simulation and then applies image-to-image translation to simulate microscopy images. We show that training an instance segmentation network on the combined synthesized samples and the initial training dataset, the test accuracy increases. The method is presented in chapter 7. Then, as a follow up research we fix the main weakness of the nucleAIzer method. In particular, instead of synthesizing instance masks using traditional parametric cell population simulation tools, we learn the distribution of objects using a generative adversarial network. We show that by learning the masks from the data, we can simulate much more realistic data (e.g. complex structures found in tissues) that is not trivial with simple parametric methods. We combine the synthesized images with the initial training dataset and show that the test accuracy increases. The method is presented in chapter 8.

Our conclusions are summarized in chapter 9.

The contributions of the author are listed in the Summary.

The connection between the thesis points and the supporting publications are listed in the Summary.

Part I

Active contour based methods

Chapter 2

Active contour segmentation with level sets

Image segmentation can be accomplished with basic methods such as thresholding, region growing/watershed techniques to name a few. Such image processing algorithms may be suitable for images where the objects can be separated easily from the background. In our case, however, images are often noisy that is caused by the side effects of the imaging pipeline. Therefore, we are interested in methods that are less prone to such effects. In microscopy image analysis, we also have some prior information about the objects to be retrieved such as their expected shape captured with shape descriptors and their approximate size. We aim to include these information into the segmentation process.

Active contour models [44] are proposed for object level segmentation of images. The basic idea is that an initial closed curve is placed in the image, ideally close to the target object, representing the segmentation by separating the plane to inner and outer regions. The curve is then evolved by minimizing an energy given as a functional. The original active contour energy [44] includes constraints of the curve itself (internal forces) and image information (external forces) often called as data term. The curve is defined using the vector valued function $\mathbf{v}(s) = \langle x(s), y(s) \rangle$ where s is the time parameter $\in [0, 1]$ of the curve. Then, the total energy of the contour is usually defined as:

$$E(\mathbf{v}(s)) = E_{int}(\mathbf{v}(s)) + E_{image}(\mathbf{v}(s)),$$

where the first term is chosen to return low energy for smoother curves, while the external term

$$E_{int}(\mathbf{v}(s)) = \alpha \int_0^1 |\mathbf{v}'(s)|^2 ds + \beta \int_0^1 |\mathbf{v}''(s)|^2 ds,$$

depends on the image and designed to return lower energy when the curve matches

the edges in the image:

$$E_{image}(\mathbf{v}(s)) = -\gamma \int_0^1 |\nabla I(x(s), y(s))|^2 ds.$$

In general form, the external force depends on some non-decreasing function of the image e.g.:

$$g(\nabla I(x)) = \frac{1}{1 + |(G_{\mu,\sigma} * \nabla I)(x)|^2}.$$

Weights α, β and γ are used to balance between the forces and G is the Gaussian filter.

This method of contour evolution is based solely on energy minimization and can can be numerically solved by explicit representation (discretization) of the curve with representative points, then the derivatives at each point are approximated using finite differences. This method however, has several drawbacks: the initial topology of the contour is assumed to be the same at the beginning and at the end of the evolution. Thus, multiple objects cannot be retrieved. Furthermore, the weights for each term must be tuned properly for success. While the topology change issue can be eliminated by smart (and complex) contour surgery techniques, *geometrical active contours* are proposed to eliminate the issue in an elegant way by defining the curve evolution from a geometrical perspective as the name suggests. Instead of relying solely on energy minimization, the geometrical approach can be viewed as solving a PDE [16, 65].

To represent the curve implicitly, *level sets* [78] are used. Let $u(t, x, y)$ is a C^2 function (the level set) and the contour ∂C is the k-th level set of u: $\partial C = \{(x, y) : u(t, x, y) = k\}\}$ and t is the time step, and usually $k = 0$. To move the curve in the normal direction, one should solve the *level set equation*:

$$\frac{\partial u}{\partial t} + V_n |\nabla u| = 0, \tag{2.1}$$

where V_n is a scalar field defines the speed at each point.

The new curve evolution equation [16] in the level set framework thus:

$$\frac{\partial u}{\partial t} = g(x)|\nabla u|\Big(\nabla \cdot \frac{\nabla u}{|\nabla u|} + v\Big).$$

In other words, the equation 2 moves the curve represented as the k-level set in the normal direction with speed depending on the mean curvature while $g(\cdot)$ is responsible to stop the motion of the contour element, when the image feature is reached, v compensates the divergence to be positive. The contour is not moved directly but the equations manipulate the level set function, which in turn moves the curve. This formulation has two great advantages over the classical energy minimiza-

tion formulation: the evolution does not depend on the particular parametrization of the curve anymore, and more importantly, the topology changes are naturally handled during the evolution of the level set as no explicit discretization of the curve is involved in the process. For a more general analysis of the contour evolution with curvature dependent motion, see the seminal work of Osher & Sethian on level sets [78] and for a general discussion about the applications of level set methods in different fields, see [77]. Another improvement over the energy-minimization based formulation is the class of *geodesic active contours* [17]. This formulation connects the energy based and geometric active contours by showing that the energy based formulation is equivalent to minimizing the length of a geodesic curve on the image in a Riemannian space. The functional is derived from variational principles (the principle of least action and Fermat-principle) and the length of the curve to be minimized is:

$$L_R = \int_0^1 g(|\nabla I(C(q))|)C'(q)dq = \int_0^{L(C)} g(|\nabla I(C(q))|)ds.$$

The resulting steepest descent equation is then embedded into the level set framework, thus the topology is handled automatically.

Although implicit representation of the contour with level sets eliminates most of the limitations of the contour tracing, it introduces its own limitations. Constructing the level set can be done by initializing it to a variant of Heaviside function, but its drawback is that it is not smooth. Other approaches are the *signed distance functions*, in particular $\phi = 1 \pm d^2$ or $\phi = \pm d$, where d is the distance from the contour, and the level set is negative in the closed region and positive in the open. It has been shown, that the level set evolution can introduce numerical errors on the level set itself and the evolution converges to unexpected curves, thus the numerical errors should be eliminated by *reinitializing* the level set periodically to signed distance function. In chapter 3 we propose such a reinitialization method that shown to be superior over competing methods.

Chapter 3

Level set maintenance with phase field based tecnhiques

3.1 Introduction

Active contours [44] have become one of the most widely-used techniques for image segmentation. These methods represent the object to be segmented as a closed curve. Early methods [44] discretized the curve to equidistant points, then moved the points by minimizing an energy functional on the curve. These implementations are often called front tracking methods in the level set literature because we track the curve based on their representative points. Front tracking methods have several difficulties. The first one is that the points should be redistributed during the contour evolution periodically. The other issue is that it is extremely challenging to handle the topology changes during the evolution. The problem of front tracking and topology management is deeply investigated in [103]. Because of the difficulties of the explicit curve representation, more recent methods [16, 65] represent the contour using an *implicit* function called the *level set* ϕ defined on $\Omega \subseteq R^n$. The contour (front) $\partial\Omega$ is represented as the k-th isocontour of ϕ e.g. $\{x|\phi(x) = k\}$, usually $k = 0$ and x lies in the inside region (Ω^-) if $\phi(x) < k$ and in the outside if $\phi(x) > k$ (Ω^+). The level set is not only capable of representing the contour at a particular time, but the whole contour evolution can be indirectly managed by manipulating the level set using the level set equation. By solving equation 2.1, we can move the implicitly represented contour in the normal direction. Level sets usually initialized to a *signed distance function (SDF)*: each point $x \in \Omega$ will get the value $d(x)$ (distance of x from the contour) adjusted with the sign depending on the location of x with respect to the contour (x is in Ω^+ or Ω^-). SDFs are preferred over the variants of the Heaviside function because of their smoothness around the contour, thus the quantities needed for the contour evolution can be approximated properly near the front if $|\nabla\phi| = 1$:

$$N = \nabla\phi \tag{3.1}$$

,

where N is the unit normal to the contour and

$$\kappa = \Delta\phi \tag{3.2}$$

,

where κ is the mean curvature. SDFs can be easily constructed algorithmically by first calculating the minimum distances

$$d(x) = \min(|x - x_C|), x_C \in \Omega \tag{3.3}$$

and then setting the sign of $d(x)$ at x depending on whether x in Ω^+ or Ω^-. A main drawback of the level set method is that during the contour evolution, the signed distance function property of the level set is not necessarily kept because of the introduction of numerical errors, and the computed quantities for updating the level set (the next contour position) become inaccurate. To ensure that the level set holds the signed distance function property, one should reinitialize the level set periodically. The SDF can be constructed using the method described above by first localizing the contour (using contour plotting in 2D or with the marching cubes algorithm in 3D) and then measuring the minimum distances. These *exact* reinitialization methods are obviously time consuming. Approximate methods also proposed that are designed to be included in the evolution equation and usually penalize the deviance from the SDF when updating the level set. These methods, however, may add instability (see [111]) and increase complexity [105]. **More importantly, these approaches may move the zero level set away from the expected stopping location**. Furthermore, depending on the application, accurate reinitialization is required for correct level set evolution. It should be noted, that since we are only interested in moving the contour, the speed function should be also computed only on the contour. The main quantities needed (normal, curvature) therefore calculated based on the level set values only in a small neighborhood around the front. This observation led to the development of *local level set* methods [3].

The Ginzburg-Landau phase field model was used in image segmentation in [86] as an alternative to the level sets. It possesses interesting advantages as greater topological freedom, the possibility of a 'neutral' initialization. Here we stress another aspect: phase field models automatically form narrow band, a useful property that can only be achieved using additive regularization [105] in the case of Hamilton-Jacobi formulation. Moreover, unlike the reaction-diffusion model [26, 111], it exhibits fast shape recovery due to the double well potential term incorporated in its functional. On the other hand, the Ginzburg-Landau phase field energy is proportional to the

length of the contour that causes curvature dependent shrinking of the level sets. In some cases they are rather destructive and the Euler's elastica is used instead (e.g. [70]).

The calculation of the fundamental quantities requires a smooth transition across a certain neighbourhood of the zero level set. On the other hand, any method dedicated to this transitional shape maintaining should have the least possible interference with the segmentation PDE. Specifically, *any curvature dependent behaviour should be an intentionally designed part of the segmentation model itself*. The Ginzburg-Landau phase field obviously violates this 'least possible interference' requirement. We propose the *balanced phase field model* [71] that eliminates the curvature driven shrinking, while maintains the smooth transition around the zero level set and show that the contour shrinking effect can be eliminated by compensating the curvature dependent motion in the phase field update equation.

3.2 The Phase Field model

Similarly to the level sets, the phase field uses an order parameter ϕ to separate two regions using an implicit real valued function over Ω. It is widely applied in materials science to model phase changes in materials, for example the solidification of fluids or crystallization of metals. In particular we are interested in the movement of the boundary (called interface in the phase field literature) between the two phases. While the interface can be modeled directly as a zero-width transition (called sharp interface model), it was shown to be not effective for example to model microstructure changes in crystals. [94]. In contrast, the phase field model, uses a diffuse interface model, that is a smooth transition between the two phases and the interface can be captured at a value half-way. This is in essence similar to level sets but unlike of them, the order parameter of the phase field is designed to be a bounded function where the inside and outside region is defined by the phase field values e.g. -1 and 1. These two phases are defined by the extreme values of a so called double-well potential. A common choice in image processing is $\phi = \frac{\phi^4}{4} - \frac{\phi^2}{2}$ with extrema $\{-1, 1\}$ 3.1.

Assuming that the used potential have extrema at values -1 and 1, if the initial phase field is initialized to values greater than 0 (inner region) and less than 0 (outer region), we minimize the Ginzburg-Landau functional to form the soft phase transition.

$$\iint_{\Omega} \gamma \left|\nabla\phi\right|^2 + \lambda \left(\frac{\phi^4}{4} - \frac{\phi^2}{2}\right) d\Omega. \tag{3.4}$$

It has two terms: the second term forces the field to form the two phases, while the first term minimizes the gradient of ϕ over Ω and it is already minimal at the

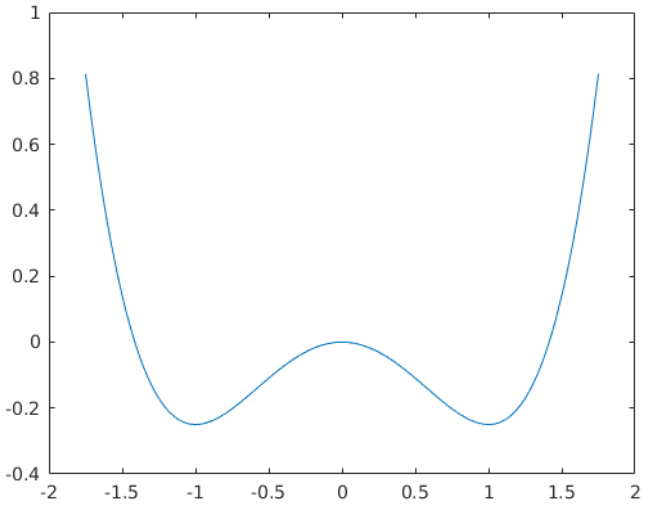


Figure 3.1: *A double well potential for the phase field model* $\frac{\phi^4}{4} - \frac{\phi^2}{2}$ *usually used in image processing.*

two phases as the potential has minimum value there, and returns minimal value when the transition is linear. This is exactly what we need: *a linear level set in a narrow band of the contour*. In the following, we investigate the phase field model for reinitialization purposes. The γ is a weight that influences the width of the transition. 3.2

3.2.1 Minimizing the contour energy using phase field

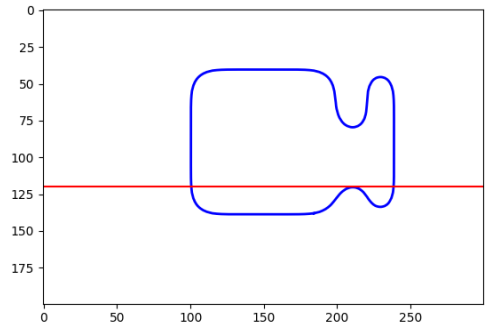
In the level set framework, the representation of contours is given by a level set function of two variables $\phi(x, y)$. The quantities of the segmentation problem are extracted from this function, such as the unit normal vector $\mathbf{n} = \frac{\nabla\phi}{|\nabla\phi|}$ or the curvature $\kappa = -\nabla \cdot \left(\frac{\nabla\phi}{|\nabla\phi|}\right)$ where ∇ is the gradient operator and "$\cdot$" stands for the scalar (dot) product, *i.e.* $\nabla \cdot \mathbf{v}$ is the divergence of the vector field $\mathbf{v}$. The level set function is usually maintained on a uniform grid and its derivatives are approximated by finite differences. Such calculation requires the level set function to be approx. linear around a small neighborhood of the zero level.

We start our analysis with the following functional:

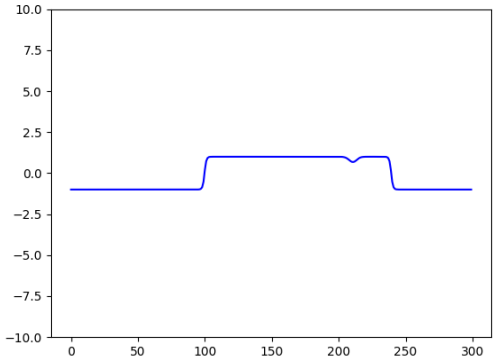
$$\iint_{\Omega} \frac{D_o}{2} |\nabla\phi|^2 + \lambda_o \left(\frac{\phi^4}{4} - \frac{\phi^2}{2}\right) dA, \quad dA = dxdy, \tag{3.5}$$

where D_o and λ_o are weights.

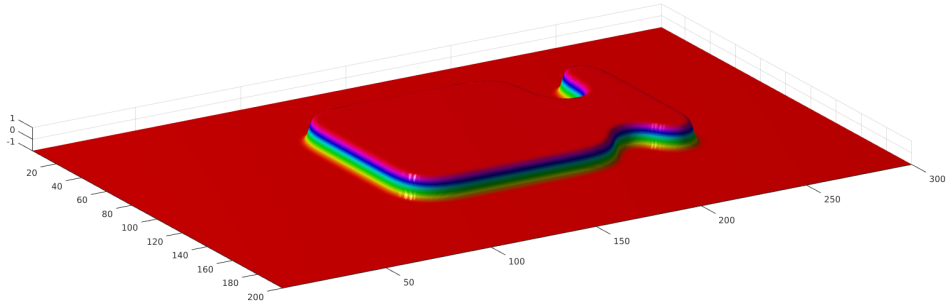
Albeit the energy (3.5) could be incorporated into any segmentation functional,



(a) *The 0-isocontour of* ϕ.



(b) *The values of* ϕ *at* $y = 50$ *(red line of 3.2a).*



(c) ϕ *on the whole* Ω *(the extreme values are* -1 *and* 1*).*

Figure 3.2: *Phase field representation of regions. Opposed to the sharp interface model, the contour is represented as a smooth transition between the two phases.*

this would extremely complicate the analysis of such a complex system. There is though another way using the phase field equation: that is using it in "shape maintaining" role, solving its associated Euler-Lagrange equation independently of and before the segmentation. In either case we wish the phase field equation ideally to maintain the shape of ϕ without moving its level sets. This idea is similar to the regularization of the level set by reinitialization or the diffusion phase of the reaction-diffusion model.

First, we assess the results of the Ginzburg-Landau phase field analysis using linear ansatz (see [86]). One can show that the width of the transition is

$$w_{o*} = \sqrt{\frac{15D_o}{\lambda_o}}, \tag{3.6}$$

and the energy of the transitional band is approximately proportional to the perimeter of the innermost (zero) level set. These approximations are valid wherever $w_{o*}|\kappa| \ll 1$. The associated Euler-Lagrange equation is

$$-D_o\triangle\phi + \lambda_o\left(\phi^3 - \phi\right) = 0\,. \tag{3.7}$$

3.3 Higher order smoothness terms for phase field model

In this section we examine a phase field $\phi(x,y)$ with Laplacian smoothness $(\triangle\phi)^2$- as a potential candidate for our purpose. Note that the origin of the energy can be chosen freely. If the phase field satisfies the condition of constancy almost everywhere except the regions of transitions, the origin is expediently chosen to be the energy level of $\phi = \pm 1$. In this case, the whole energy is equivalent to the energy of the transitions and can be written as

$$\iint_\Omega \frac{D}{2}\left(\triangle\phi\right)^2 + \lambda\left(\frac{\phi^4}{4} - \frac{\phi^2}{2} + \frac{1}{4}\right)dxdy\,. \tag{3.8}$$

The Euler-Lagrange equation associated with this functional is

$$D\triangle\triangle\phi + \lambda\left(\phi^3 - \phi\right) = 0\,. \tag{3.9}$$

To estimate the energy (3.8) we use curvilinear coordinates.

3.3.1 Approximate system energy

In the vicinity of the curve $\mathbf{r}(s)$ (s is the arc length parameter), the plane can be parameterized as $\mathbf{R}(s,p) = \mathbf{r}(s) + p\mathbf{n}(s)$, where $\mathbf{n}(s)$ is the unit normal vector of the curve at s and p is the coordinate in the normal direction. The metric tensor

components are the scalar (dot) products of the covariant basis vectors $\mathbf{R}_s = \frac{\partial \mathbf{R}}{\partial s}$, $\mathbf{R}_p = \frac{\partial \mathbf{R}}{\partial p}$ and takes the form $[g_{ik}] = \operatorname{diag}\left[(1-p\kappa)^2, 1\right]$, where $\kappa = \kappa(s)$ is the curvature of the curve $\mathbf{r}$ at s given by the Frenet-Serret formula $\mathbf{n}_s = -\kappa \mathbf{e}$ ($\mathbf{e}$ is the unit tangent vector). The invariant infinitesimal area is $dA = \sqrt{g} ds dt$ where $g = \det[g_{ik}]$. Using these, the Laplacian $\triangle\phi$ in the curved system $(u^1, u^2) = (s, p)$ is given by the Laplace-Beltrami operator $\triangle\phi = \frac{1}{\sqrt{g}} \frac{\partial \sqrt{g} g^{ik} \frac{\partial \phi}{\partial u^k}}{\partial u^i} = \frac{1}{1-p\kappa} \left[\frac{\partial (1-p\kappa)^{-1} \frac{\partial \phi}{\partial s}}{\partial s} + \frac{\partial (1-p\kappa) \frac{\partial \phi}{\partial p}}{\partial p} \right]$ (in the general expression the components of the inverse metric $\left[g^{ik}\right] = [g_{ik}]^{-1}$ and the Einstein summation convention are used). It can be rearranged as

$$\triangle\phi = \frac{1}{(1-p\kappa)^2} \frac{\partial^2 \phi}{\partial s^2} + \frac{p}{(1-p\kappa)^3} \frac{d\kappa}{ds} \frac{\partial \phi}{\partial s} + \frac{\partial^2 \phi}{\partial p^2} - \frac{\kappa}{1-p\kappa} \frac{\partial \phi}{\partial p}. \tag{3.10}$$

Now we choose $\mathbf{r}(s)$ to be the zero level set and use the following simplifications:

1. the constant level sets are equidistant to $\mathbf{r}(s)$ *i.e.* $\phi = const \rightarrow \frac{\partial^n \phi}{\partial s^k \partial p^{n-k}} = 0$, $k \in [1, n]$ along the parameter lines $p = const$
2. the transition is confined to a stripe $\left(-\frac{w}{2}, \frac{w}{2}\right)$ along the zero level set contour
3. the osculating circle is significantly bigger than the stripe width: $1 - p\kappa \approx 1$

then energy (3.8) expressed in the (s, p) system becomes

$$\oint \int_{-\frac{w}{2}}^{\frac{w}{2}} \frac{D}{2} \left(\phi'' - \kappa\phi'\right)^2 + \lambda \left(\frac{\phi^4}{4} - \frac{\phi^2}{2} + \frac{1}{4} \right) dp ds \tag{3.11}$$

$\phi(s, p) = \phi(p)$ and $\kappa = \kappa(s)$ is the curvature measured on the zero level set and prime notation is used for the derivatives wrt p.

3.3.2 Cubic ansatz

In the presence of the second derivative ϕ'' in (3.11), the linear ansatz is not applicable. The next simplest choice is a cubic ansatz with boundary conditions $\phi\left(-\frac{w}{2}\right) = -1$, $\phi\left(\frac{w}{2}\right) = 1$ and $\phi'\left(-\frac{w}{2}\right) = \phi'\left(\frac{w}{2}\right) = 0$. The function satisfying these conditions is

$$\phi(p) = -\frac{4}{w^3} p^3 + \frac{3}{w} p. \tag{3.12}$$

Its derivatives are:

$$\phi' = -\frac{12}{w^3} p^2 + \frac{3}{w}, \quad \phi'' = -\frac{24}{w^3} p, \quad \phi''' = -\frac{24}{w^3}. \tag{3.13}$$

The square of the approximate Laplacian, obtained from (3.10) is $\left(\phi'' - \kappa\phi'\right)^2 = \left[-\frac{24}{w^3} p - \kappa\left(-\frac{12}{w^3} p^2 + \frac{3}{w}\right)\right]^2$. The inner integral in (3.11) is symmetrical, therefore

the terms having odd powers of p do not contribute to the energy. Integrating this smoothness term results $\frac{24D}{w}\left(\frac{1}{w^2}+\frac{\kappa^2}{10}\right)$. Now it can be seen that the appearance of the curvature in the energy violates the assumptions 1 and 2 given in section 3.3.1. The contribution of the second term $\frac{\kappa^2}{10}=\frac{1}{10r_O^2}\ll\frac{1}{w^2}$ is however, very modest thus omitted in the subsequent calculations. Similarly, the inner integral of the phase field double well potential term $\lambda\left(\frac{\phi^4}{4}-\frac{\phi^2}{2}+\frac{1}{4}\right)$ is approximately $0.1\lambda w$, hence the approximate energy of (3.11) is $L\left(\frac{24D}{w^3}+\frac{\lambda w}{10}\right)$ (L is the contour length). Deriving it wrt w, the optimal width of the transitional region is

$$w_*=\sqrt[4]{\frac{720D}{\lambda}}\,. \tag{3.14}$$

3.3.3 The motion of the level sets

The Euler-Lagrange equation (3.9) can be expressed in the curvilinear system aligned with the zero level set applying the Laplace-Beltrami operator once again to the equation (3.10) (and multiplying the result by $\sqrt{g}=1-p\kappa$). The Euler-Lagrange terms having the derivatives of ϕ by the contour parameter s can be omitted in the result. This approximate equation is

$$\begin{gathered}D\left(-A\phi'-2\kappa\phi'''-\kappa^2\phi''+\phi''''\right)+\lambda\left(\phi^3-\phi\right)=0\\ A=3p\left(\frac{d\kappa}{ds}\right)^2+\frac{d^2\kappa}{ds^2}+\kappa^3\,.\end{gathered} \tag{3.15}$$

The shape of the numerical solution for (3.15) is close to the cubic ansatz (see Fig. 3.4). Due to the assumed symmetry of the zero level set, its motion is governed by $-D\left(\frac{d^2\kappa}{ds^2}+\kappa^3\right)\phi'-2D\kappa\phi'''=0$ or using the cubic ansatz at $p=0$ and (3.14):

$$\frac{48D}{w_*^3}\kappa-\left(\frac{d^2\kappa}{ds^2}+\kappa^3\right)\frac{3D}{w_*}=0\,. \tag{3.16}$$

Equation (3.16) describes either a static state wherever the curvature is identically zero or shrinking proportional to the curvature where the radius r_O of the osculating circle is significantly bigger than the thickness of the transition and this width varies slowly.

For contours with constant curvature a static solution would be at radius $\frac{48\kappa}{w_*^3}=\frac{3\kappa^3}{w_*}\rightarrow r=\frac{w_*}{4}$, however this is not the case. Around this curvature value neither the assumption $1-p\kappa\approx 1$ nor the cubic ansatz approximation are valid. Under these circumstances the phase field function can no longer be modeled with cubic ansatz (see Fig. 3.4 right column). The theoretical minimum value while the phase field is shape-retaining is $\frac{w_*}{2}$.

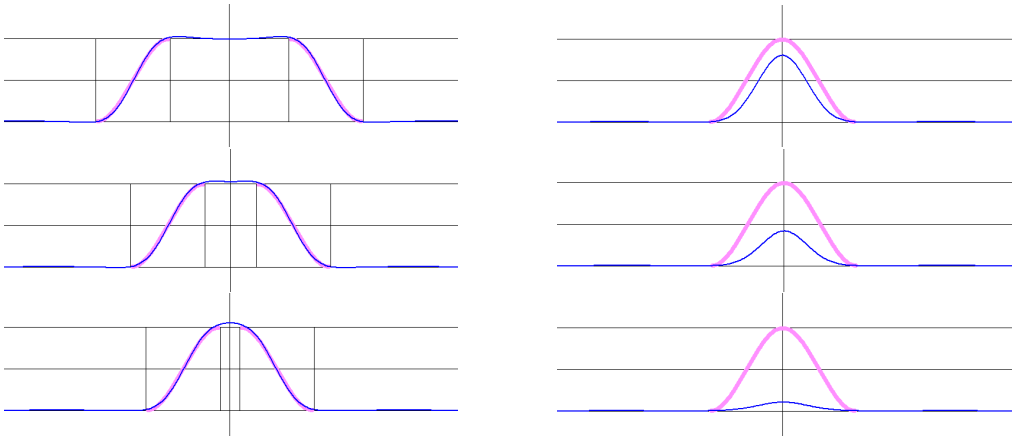


Figure 3.3: *Alteration of the phase field function in normal direction (thin blue line). Left column: small curvature, the cubic ansatz (thick violet line) is valid. Right column: high curvature, cubic approximation is invalid.*

Therefore, we conclude that energy (3.8) does not fulfill our expectation stated in the beginning of this section because it still has a curvature dependent term.

3.3.4 Motion of the level sets of the original model

Assuming again that the conditions that led to the simplified energy expression (3.11) are valid, the Euler-Lagrange equation of the Ginzburg-Landau phase field (3.7) reduced to the normal direction using the Laplacian (3.10) at $1 - p\kappa \approx 1$ and $\frac{\partial^n \phi}{\partial s^n} = 0$ is: $-D_o\left(\phi'' - \kappa\phi'\right) + \lambda_o\left(\phi^3 - \phi\right) = 0$. From this expression, the motion of the zero level set is governed by

$$D_o \kappa \phi' = 0, \tag{3.17}$$

that is a pure curvature-driven motion.

3.4 Phase field model for reinitialization purpose

The motion of the zero level set is basically curvature driven both for the Ginzburg-Landau (3.17) and the higher order smoothness (3.16) models. This effect can be eliminated by the appropriate combination of the smoothness terms $(\nabla\phi)^2$ and $(\triangle\phi)^2$. First we calculate the optimal width for the functional

$$\iint_\Omega \frac{D}{2}\left|\triangle\phi\right|^2 - \frac{D_o}{2}\left|\nabla\phi\right|^2 + \lambda\left(\frac{\phi^4}{4} - \frac{\phi^2}{2} + \frac{1}{4}\right) dA\,. \tag{3.18}$$

The approximate energy - using simplifications 1.-3. introduced in section 3.3.1 - is

$$L\int_{-\frac{w}{2}}^{\frac{w}{2}} \frac{D}{2}\left(\phi'' + \kappa\phi'\right)^2 - \frac{D_o}{2}\left(\phi'\right)^2 + \lambda\left(\frac{\phi^4}{4} - \frac{\phi^2}{2} + \frac{1}{4}\right) dp, \tag{3.19}$$

where the length of the contour is $L = \oint ds$ is independent of w. Substituting the cubic ansatz (3.12), the integral (3.19) (divided by L) becomes

$$24D\left(\frac{1}{w^3} + \frac{\kappa^2}{10w}\right) - \frac{12D_o}{5w} + \frac{\lambda w}{10}\,. \tag{3.20}$$

The term dependent on the square of the curvature is again negligible, hence omitted. From expression (3.20), the optimal width is given by derivation wrt w

$$\boxed{\lambda w^4 - 24D_o w^2 - 720D = 0} \tag{3.21}$$

that can be solved for the optimal width w_*. The solution is

$$w_* = \sqrt{\frac{12}{\lambda}\left(-D_o + \sqrt{D_o^2 + 5D\lambda}\right)}\,. \tag{3.22}$$

Now we use the approximate Euler-Lagrange equation associated with (3.19)

$$D\left(-A\phi' - 2\kappa\phi''' - \kappa^2\phi'' + \phi''''\right) + D_o\left(\phi'' - \kappa\phi'\right) + \lambda\left(\phi^3 - \phi\right) = 0 \tag{3.23}$$

to derive condition for the curvature-independent solution (here A is defined in (3.15)). From (3.23) the curvature-dependent term is eliminated with the condition: $-D_o\phi' - 2D\phi''' \doteq 0$. Substituting the cubic ansatz (3.12) (at $p = 0$) we get:

$$\boxed{-D_o\frac{3}{w} + D\frac{48}{w^3} = 0} \;\rightarrow\; D = \frac{w^2}{16D_o}\,. \tag{3.24}$$

The width (3.21) and the curvature (3.24) constraints determine the weights for the solution with curvature driven shrinking effect removed. There are other terms, e.g. terms included in factor A, but the influence of those is much weaker. In fact the impact of the term $D\left(\frac{d^2\kappa}{ds^2} + \kappa^3\right)$ is similar to that of the solution of the Euler's elastica $\oint \frac{D}{2}\kappa^2 ds$ with associated Euler-Lagrange equation: $D\left(\frac{d^2\kappa}{ds^2} + \frac{1}{2}\kappa^3\right) = 0$. The numerical tests confirm that the phase field used in this manner - satisfying equations (3.21), (3.24) - essentially fulfills the "transitional shape maintenance" role while standing still.

3.4.1 Determining weights

Given two constraints (3.21), (3.24) for the energy (3.18), one of the weights can be chosen freely (say $D_o = 1$). The calculation of the remaining weights are as follows. First determine the width: depending on the highest order of the derivatives n (occurring either in the segmentation model or the phase field itself), we need at least $n + 1$ grid points around the zero level set using finite central difference schemes. This suggests about twice as big (as a cautious choice) thickness of the phase field transition to remain within the range where it is approximately linear, *i.e.* $w \geqq 2\,(n+1)$ is recommended. Second, solving (3.21) and (3.24), the weights the functions of the width parameter w such as:

$$D_o = 1,\ D = \frac{w^2}{16},\ \lambda = \frac{21}{w^2}\,. \tag{3.25}$$

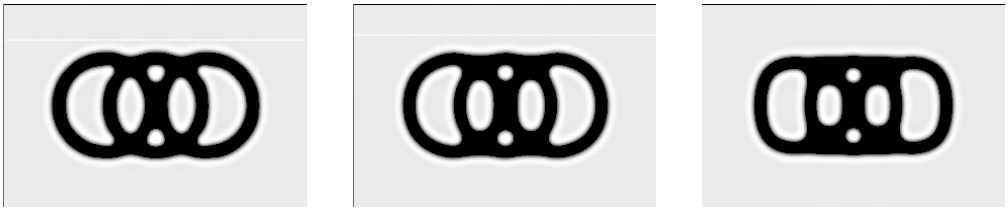
The Euler-Lagrange equation associated with the proposed energy (3.18), using the calculated weights (3.25) dependent on the width parameter w is therefore

$$\boxed{\frac{w^2}{16}\triangle\triangle\phi + \triangle\phi + \frac{21}{w^2}\left(\phi^3 - \phi\right) = 0}\,. \tag{3.26}$$

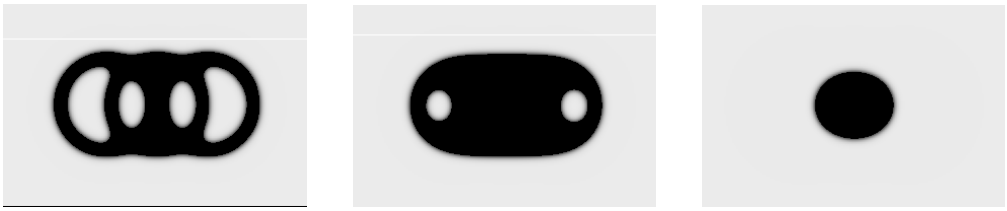
In (3.26) the Laplace operator can be expressed wrt the standard basis as $\triangle\Phi = \frac{\partial^2\Phi}{\partial x^2} + \frac{\partial^2\Phi}{\partial y^2}$, $\Phi \in \{\phi, \triangle\phi\}$ and discretized on a uniform grid using finite differences. Its gradient descent was used in the tests. The method can be efficiently implemented as a 5×5 linear filter plus a point-wise cubic term acting on the uniform grid used to discretize the level set function. The approximation $r_{min} \approx X\,(n+1)$ (where X is the grid size that can be smaller or greater than a pixel) also determines the size of the segmentable smallest image-feature.

3.5 Experimental Evaluation

In this section, we show that our balanced phase field model a) maintains a smooth transition of the level set in a narrow band during the evolution, while b) it has minimal side effect on the contour at the same time. This section is organized as follows: first, we compare the Ginzburg-Landau phase field model to the proposed one to show that the latter has much better contour preserving performance. Next, we compare the proposed model, the reaction diffusion model [111] and a reinitialization method [80] on synthetic and real data.



(a) *The balanced phase field - interfaces are barely moved.*



(b) *The Ginzburg-Landau phase field: interfaces are displaced significantly.*

Figure 3.4: *Alteration after the same iteration numbers of the phase fields when* $w = 10$*. Weights are set according to (3.25) and (3.6) for the balanced model and the Ginzburg-Landau model respectively.*

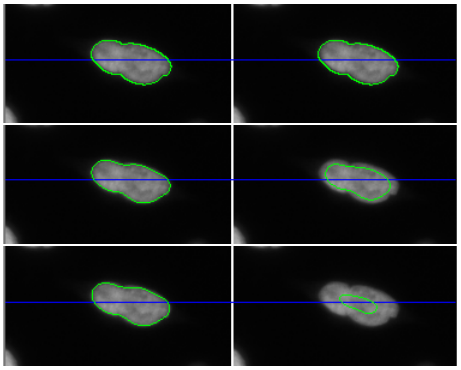
3.5.1 Stability tests: comparing to the Ginzburg-Landau model

The test environment is prepared to guarantee the synchronous snapshot production for the illustrations: the simulation space is splitted in the middle such that the phase field evolutions are governed by the proposed energy (3.18) on one side and the Ginzburg-Landau energy (3.5) on the other.

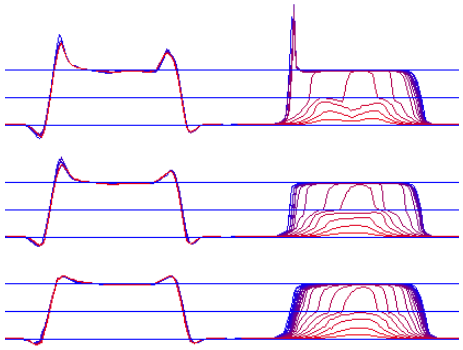
For the first test, the pure phase field equations were used. Fig. 3.4 left shows the initial contour preserving capability of the balanced phase field model compared with the Ginzburg-Landau model on the right side.

The proposed phase field model was also applied to real data segmentation, using a selective segmentation model [70]. The energy to be minimized is: $E = \alpha\mathcal{S} + \beta\mathcal{P} + \gamma\mathcal{D} + \delta\mathcal{E}$, where $\mathcal{P} = \frac{1}{2}\left[\oint dA - q\left(\oint ds\right)^2\right]^2$ is the "plasma shape" prior (q is the shape parameter, the ratio of the enclosed area and the square of the perimeter), the data term of the original model was replaced by the simplest anisotropic edge energy $\mathcal{D} = \oint \nabla I \cdot \mathbf{n} ds$ (see [46]); the $\mathcal{E}$ is the Euler elastica, while α, β, γ, δ are weights. This segmentation model was chosen, because of its sensitivity to any size decreasing effect due to the term $\mathcal{S} = \frac{1}{3}\left(\oint dA - A_0\right)^3$ which is used at its inflection point at the preferred size A_0. The initial contours were produced by simple thresholding. Segmentation steps and the sequence of the phase field sectional values along a horizontal line are shown in Fig. 3.5.

For the test $X = 1$ pixel grid size and $w = 10$ width values were used; the maximum speed of the evolution by the segmentation model was set such that its



(a) *Segmentation result.*



(b) *Reinitializing in iterations 2, 5, 15 for shape maintenance.*

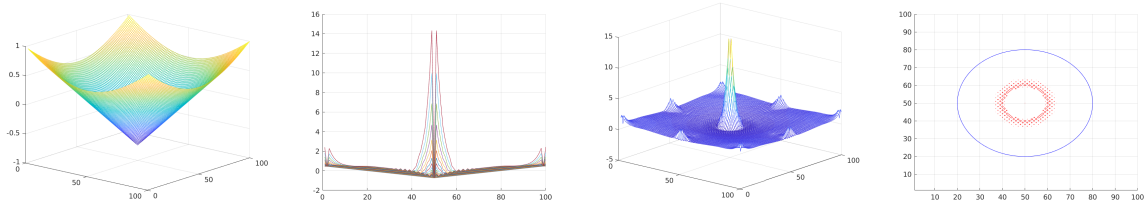
Figure 3.5: *Evolution of the selective segmentation example (grid size: 1 pixel, w=10). First columns: the balanced phase field model; Second columns: the Ginzburg-Landau phase field models.*

maximal value could not exceed the grid size. Preceding the segmentation step, the gradient descent equation of (3.26) is iterated and the phase field is updated in a reinitialization loop to recover a reasonably smooth interface. Then the segmentation gradient descent moves the contour towards the solution (but deteriorates its shape).

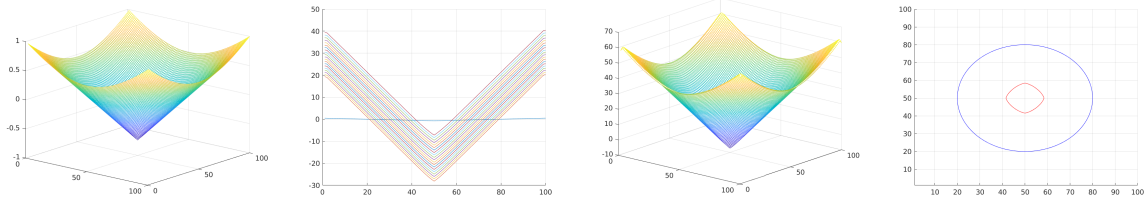
The transitional-shape recovery and the segmentation results using the balanced and the Ginzburg-Landau models, depending on the number of the phase field iterations in the reinitialization step (denoted by n) are assessed here: At $n = 2$ neither the balanced nor the Ginzburg-Landau models can be considered stable, at $n = 5$ both models provide stable transition, however the Ginzburg-Landau model develops extremely steep slopes, while at $n = 15$, both models exhibit high degree of stability as well as widths close to the designed/predicted ones. Regardless the number of the phase field iterations used, the selective segmentation [70] combined with the Ginzburg-Landau model ends up in the collapse of the contour, whilst its combination with the balanced phase field model provides the expected solution.

3.5.2 Comparing to the Reaction-Diffusion model

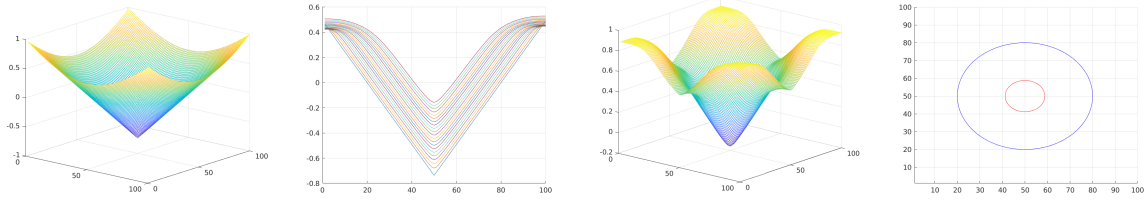
The reaction diffusion model (RD) [111] is also proposed to diminish the interference with the segmentation (active) model. The shape maintenance of the level set function is achieved by adding a diffusion term $\varepsilon\Delta\phi$ to the gradient descent of the active model, therefore $\phi_t = \varepsilon\Delta\phi + \frac{1}{\varepsilon}F|\nabla\phi|$, where ε is a small constant and F represents the gradient descent equation of the active model. We first show that both the RD and the proposed balanced phase field model fulfill the shape maintenance role. Next, we show that RD moves the interface more significantly compared to the proposed



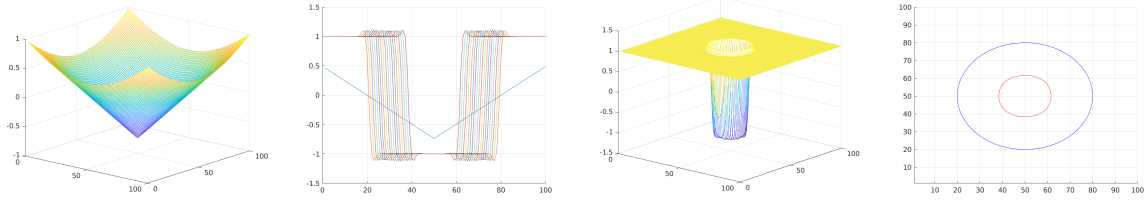
(a) *Direct implementation (no reinitialization e.g. simple Ginzburg-Landau model.). (DI)*



(b) *Reinitialization method. (RM)*



(c) *Reaction-Diffusion model. (RD)*



(d) *The balanced phase field model. (BPF)*

Figure 3.6: *Comparing different reintitialization methods. Columns from left to right: initial level sets; second: level set during evolution (intersection at $x = 50$); third: final level sets, fourth: final contours.*

	Synthetic			Real		
	RM	RD	BPF	RM	RD	BPF
result mean	0.943	0.339	0.882	0.365	0.093	0.343
peak mean	0.948	0.983	0.975	0.608	0.802	0.794

Table 3.1: *Segmentation accuracy of the synthetic and the real tests. Peak: the best scores reached during the evolution. RD can not keep the accuracy of the active model in long term, it has massive side-effects while the BPF behaves similarly to the reinitialization method.*

model. In case of RD, this shrinking side-effect eventually leads to the disappearing of some objects. For the quantitative results, we borrowed a Jaccard-distance based metric similar to the one used in the 2018 Data Science Bowl (DSB2018) competition. [1] The only modification is that we used the threshold levels $t = 0.1$ to $t = 0.95$ with steps 0.5 (inclusive).

Shape maintenance tests:

We compared three different models to ours. We tested the reaction diffusion method (RD), the balanced phase field model (BPF), a reinitialization method (RM) [80] using the reinitialization equation: $\phi_t + S(\phi_0)(|\nabla\phi| - 1) = 0$, where $S(\phi) = \frac{\phi}{\sqrt{\phi^2 + (|\nabla\phi|\Delta x)^2}}$, (the same method as the one used in the RD paper and referred to as re-initialization. (For the implementation, see the online supplementary material of that paper), and lastly, no shape maintenance) (DI - direct implementation). The first test inherits from Fig. 5 of the RD paper [111], $\Delta t_1, \Delta t_2$ (used for the numerical solution of the RD equation, see the RD paper for details) are 0.1. For the RM, Δx and Δy is 1 and $\alpha = 0.5$, while $w = 8$ in the BPF model. The force term in the gradient descent of the active contour model is simply 1, grid dimensions: 100×100, number of iterations: 200. The results are shown in Fig. 3.6.

Synthetic tests:

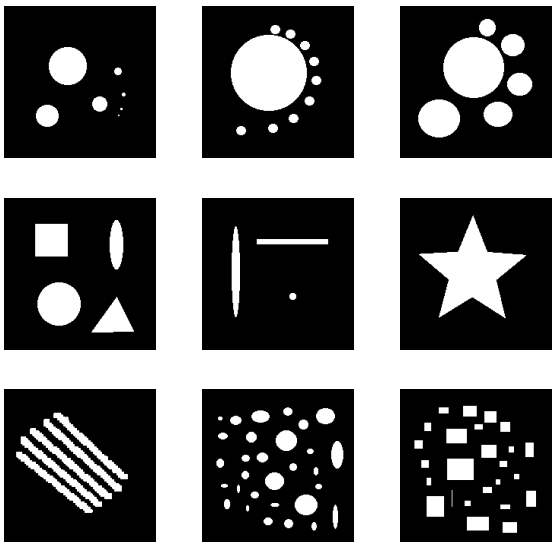
The models are compared to each other by performing a level set evolution using 11 synthetic initial contours (subset of these masks are shown in Fig. 3.7) using 0 as a force term in the active model. In this setting, we would assume that the initial contours are not moving. A sample evolution is visualized in Fig. 3.7 using the RM, RD and the BPF methods. The same test performed with all of the synthetic initial contours. The quantitative results using the modified DSB2018 metric presented in Fig. 3.8 left. The ground truth is the initial contour and the accuracy is measured during the evolution on every image. Simple statistics summarizes the results in Table 3.1 left.

Real tests:

We also compared the methods to each other using the geodesic active contour model (GAC) [17] on 51 real images containing nuclei extracted from the DSB2018 training set with random sampling. In the GAC model, the force term in the gradient descent equation is: $\nabla \cdot \left(g(I) \frac{\nabla \phi}{|\nabla \phi|} \right) + \nu g(I)$, where $g(I) = \frac{1}{1+(\nabla G(15,1.5) * I)^2}$ is the edge indicator function (the same as the one used in the RD paper for the tests). The quantitative results with this model are presented in Fig. 3.8 right. The parameters are unchanged. A sample test image used for this test is shown in Fig. 3.9. Simple statistics presented in Table 3.1 right. The parameters left unchanged since the last test, except the Δt_2 that is 0.001 in this case. In conclusion, the BPF outperformed RD both for the synthetic and real tests and produced results that are comparable to the RM method.

3.6 Summary

In this chapter we proposed and analyzed a *balanced phase field model* as an alternative to the Ginzburg-Landau level set framework. The proposed model exhibits very fast shape recovery (essentially) without moving the level sets *i.e.* its interference with the "active" (*e.g.* segmentation) PDE is negligible. This important property makes this level set formulation suitable for accurate segmentation. Similar balancing could be used for any model that includes Laplacian smoothness term in their gradient descent equation such as the reaction-diffusion model.



(a) *Test images for the synthetic tests.*

(b) *Evolution with the reinitialization method. (RM)*

(c) *Evolution with the reaction diffusion model. The model has serious side effect on the contours causing that several small objects melt to big ones, then completely disappear. (RD)*

(d) *Evolution using the balanced phase field model. (BPF)*

Figure 3.7: *Left: sample synthetic masks encoding initial contours; right: level set evolutions on a sample image from the left side with RM/RD/BPF from top to bottom, respectively.*

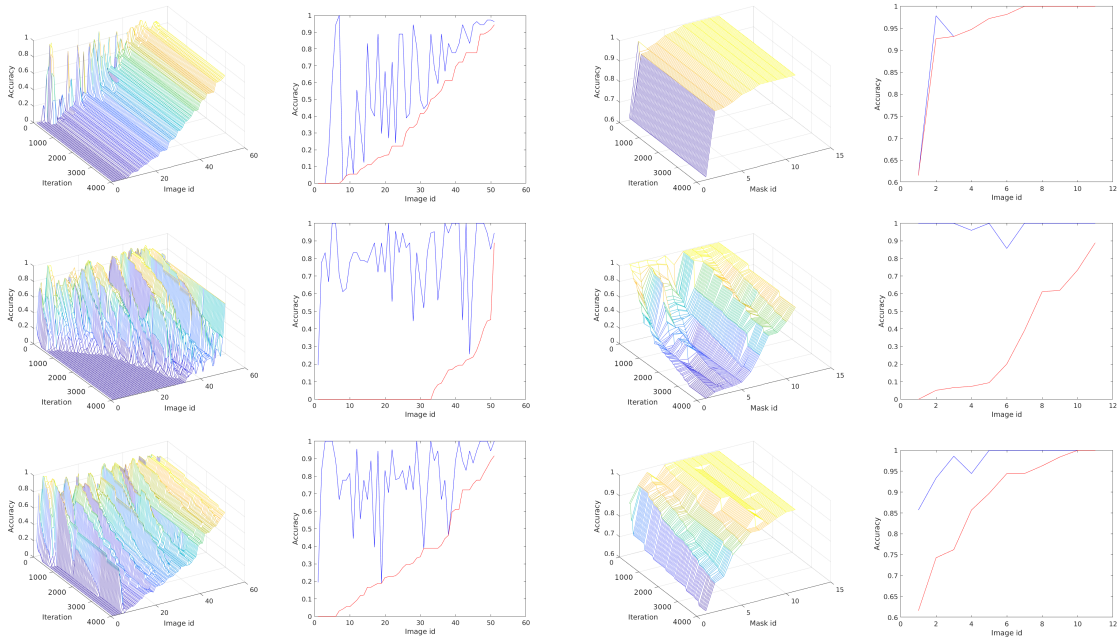
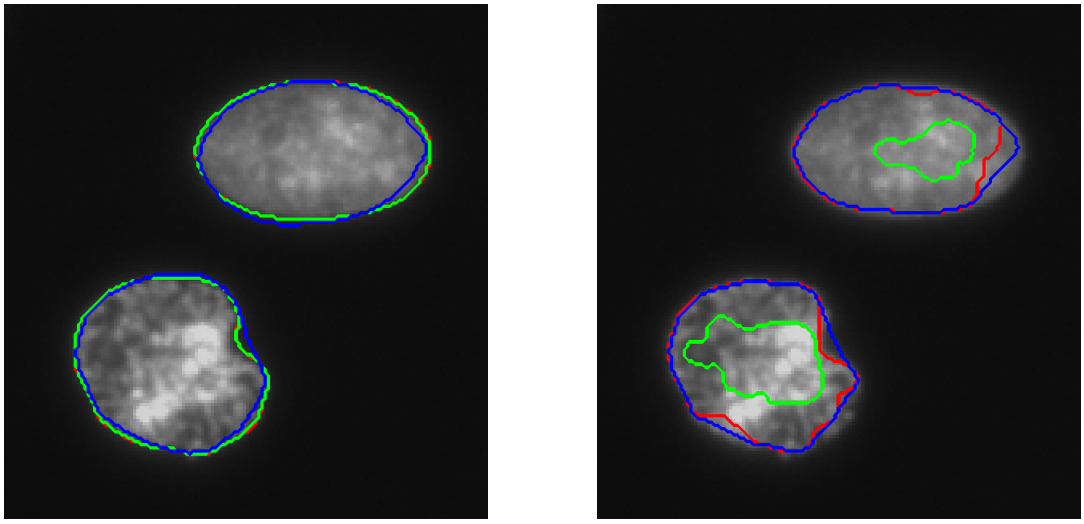


Figure 3.8: *Left: quantitative results for the synthetic test. From top to bottom: RM, RD, BPF. Columns: first: accuracy on each image during the evolution, second: peak (maximum acc. on an image, blue) and the final acc. for the images. Right: results on patches extracted from the DSB2018 training set, GAC model ($\nu = 0.5$). From top to bottom: RM, RD, BPF.*



(a) *The most accurate contours achievable with each of the methods.*

(b) *The contours after 40k iterations with different reinitialization methods.*

Figure 3.9: *Active contour evolutions with different reinitialization methods on a patch from the DSB2018 dataset: red: BPF, green: RD, blue: RM. Even if the active model is able to achieve good accuracy (Fig. 3.9a) the contour vanishes by the time if we use the RD method (*serious side effect *on the active model). The proposed model has contour preserving ability comparable to the reference method (*marginal side effect *on the active model) (Fig. 3.9b).*

Chapter 4

A selective active contour model in 3D

4.1 Introduction

We approach the problem of segmenting single cells sharing similar morphologies using the 3D extension of the active contour framework [61, 110]. In [42], Molnar et al. introduced a family of shape descriptors to capture objects with predefined area, contour length, second moment, curvature or arbitrary combination of these. This model was able to precisely capture circular, elliptical and amoeba-like cell shapes. In this chapter we present novel energies capable to segment 3D objects satisfying certain shape criteria. Our method [72] is based on minimizing functionals designed to penalize shapes unfit to given parameters. We present a shape prior which forces objects to assume a prescribed surface/volume ratio. Using this prior it was possible to segment objects satisfying only the given parameters. This selective capability of the model is demonstrated on synthetic and fluorescent microscopy images.

The outline of this chapter is the following. First, we briefly review the notations and conventions in Section 4.2. The energy functionals used for segmentation are constructed in Section 4.3, for which the corresponding Euler-Lagrange and level set equations are given in Section 4.4. The tests are carried out in Section 4.5, while we discuss the results and future work in Section 4.6.

4.2 Notations and conventions

Surfaces will be denoted by $\mathbf{S} \subseteq \mathbb{R}^3$ or $\mathbf{S}(u, v) \in \mathbb{R}^3$ in parametrized form, where u and v are surface parameters. $\mathbf{S}_u, \mathbf{S}_v \in \mathbb{R}^3$ will denote the partial derivatives in the tangent space at $\mathbf{S}(u, v)$. Recall that $\mathbf{S}_u \times \mathbf{S}_v$ is normal to the surface. The inward pointing unit normal $\frac{\mathbf{S}_u \times \mathbf{S}_v}{|\mathbf{S}_u \times \mathbf{S}_v|}$ is denoted by $\mathbf{n}$. The sum curvature of the surface is denoted by K, while K_G is the Gaussian curvature. The integral $\int dS =$

$\int \sqrt{|\mathbf{S}_u|^2|\mathbf{S}_v|^2 - (\mathbf{S}_u \cdot \mathbf{S}_v)^2}dudv$ gives the surface area and $\int dV = -\frac{1}{3}\int \mathbf{S} \cdot (\mathbf{S}_u \times \mathbf{S}_v)dudv$ gives the volume of a surface $\mathbf{S}$, where dS and dV are the surface and volume element respectively.

To handle topological changes in the evolving surfaces, a level set representation is used. In this chapter, level set functions are denoted by $\phi = \phi(t, \mathbf{x})$, where $t \in \mathbb{R}$ and $\mathbf{x} = (x_1, x_2, x_3) \in \mathbb{R}^3$ are the time and space variables respectively. According to this, ϕ_t denotes the partial derivative with respect to the time and $\nabla\phi$ denotes the spatial gradient $\nabla\phi = (\phi_{x_1}, \phi_{x_2}, \phi_{x_3})$. The Hessian matrix of ϕ is denoted by $\mathbf{H}(\phi) = (\phi_{x_i x_j})_{1\leq i,j\leq 3}$.

4.3 Functionals for selective object segmentation

In this section we present the functionals used for selective object segmentation.

4.3.1 Size priors

Our functionals responsible for the size of the object will serve two purposes. The *minimizer* functional will force the surface to shrink, while the *ratio* functional will force it to assume a prescribed volume. For minimizer, we use

$$\mathcal{M}_V(\mathbf{S}) = \int dV = -\frac{1}{3}\int \mathbf{S} \cdot (\mathbf{S}_u \times \mathbf{S}_v)dudv, \tag{4.1}$$

which minimizes volume. The functional meant to force the objects towards certain volume is defined by

$$\mathcal{R}_V(\mathbf{S}) = \frac{1}{kV_0^k}\left(\int dV - V_0\right)^k, \tag{4.2}$$

where V_0 is the preferred volume and $k \in \mathbb{N}$ is arbitrary. We refer to (4.2) as the volume prior. For even k, the functional prefers objects with volume V_0. For odd k it has an inflection at V_0, thus while it prefers zero volume, it has no effect on objects with volume V_0.

4.3.2 Shape prior

To control the shape of the objects, we have to use a functional which penalizes deviation from a given shape descriptor. For this purpose, we define a prior, which was designed to prefer shapes with given surface/volume ratio and is defined by

$$\mathcal{S}_\mathcal{A}(\mathbf{S}) = \frac{1}{2V_0^2}\left[\left(\int dS\right)^{\frac{3}{2}} - p\int dV\right]^2, \tag{4.3}$$

where p is a fixed parameter and S_0 is the preferred surface. The functional penalizes the deviation from the ratio $p = \frac{surface^{\frac{3}{2}}}{volume}$. From now on, we refer to (4.3) as the amoeba prior.

The minimal value of the amoeba parameter is $p = 3\sqrt{4\pi} \approx 10.6$, which is achieved only by spheres. Thus by setting the amoeba parameter to this value, we can force objects towards spherical forms (Fig. 4.1).

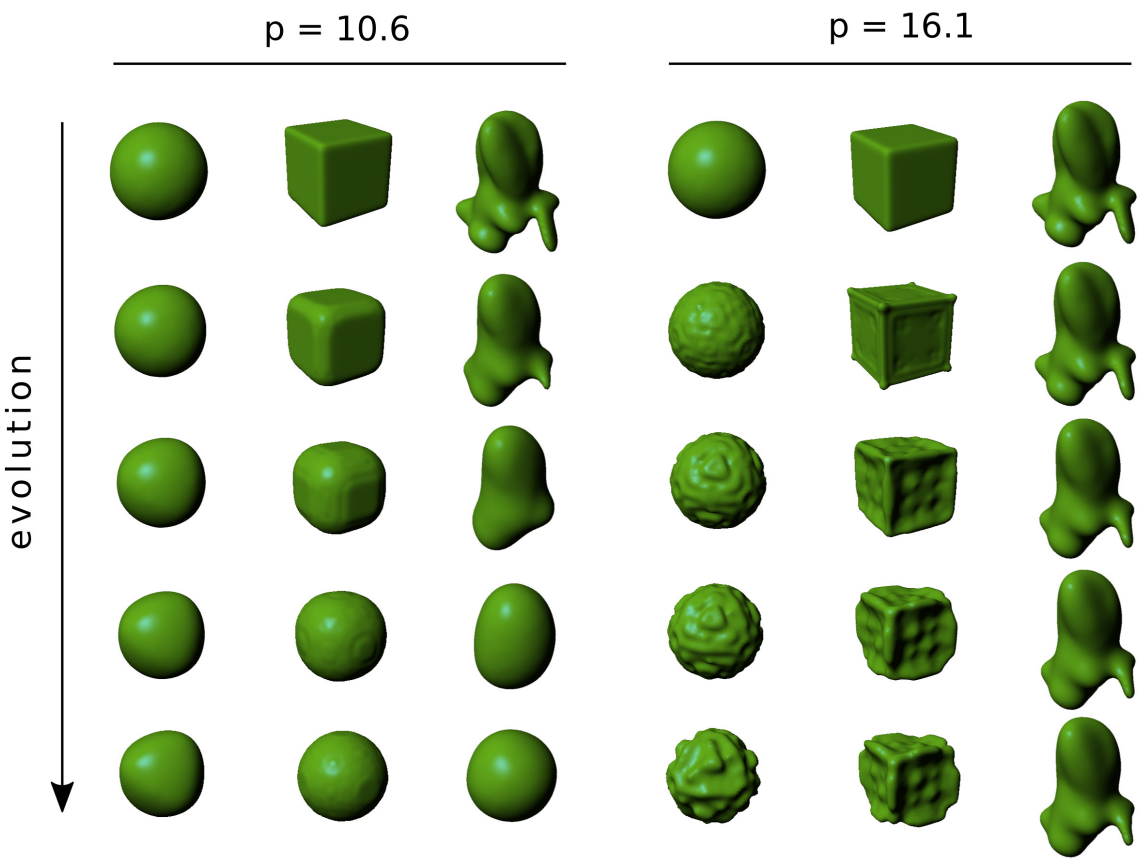


Figure 4.1: *Effects of the amoeba prior on various initial objects.*

4.3.3 Smoothness terms

In order to avoid slow convergence or unstable behavior caused by the shape and size priors, additional functionals can be introduced to regulate the solution. In general, when higher-order derivatives are not present in the equations used for segmentation, simple smoothness terms, for example surface or volume minimizers can be used (see Section 4.3.1). Because of interference with other terms, these will not work in our case. Instead we use the so-called *Euler elastica*, which measures

the bending energy of the surface. In two dimensions, it has been applied e. g. to impainting problems [19]. For a general survey on Euler elastica, see [73]. The functional itself is defined by

$$\mathcal{E}(\mathbf{S}) = \frac{1}{2}\int K^2 dS, \tag{4.4}$$

where K is the sum curvature of the surface. It is worth to note that this is a dimensionless quantity.

4.3.4 Data term

During the tests, we used the anisotropic data term (see [45])

$$\mathcal{D}(\mathbf{S}) = \int \nabla I \cdot \mathbf{n} dS, \tag{4.5}$$

but in principle, a large range of data terms are feasible.

4.3.5 The composite functional for selective segmentation

In practice, the selective segmentation consists of two steps. First we find the individual objects in the image, then we turn on the shape and size priors for each object if the presegmentation had converged. Alternatively, the active surfaces can be initialized by thresholding. The two functionals we use are

$$\mathcal{L} = \alpha\mathcal{D} + \beta\mathcal{M}_V \tag{4.6}$$

and

$$\mathcal{L} = \alpha\mathcal{D} + \beta\mathcal{S}_\mathcal{A} + \gamma\mathcal{R}_V + \delta\mathcal{E} \tag{4.7}$$

where $\mathcal{S}_\mathcal{A}$ is the amoeba prior, $\mathcal{M}_V$ is the volume minimizing functional, $\mathcal{R}_V$ is the volume prior and $\mathcal{E}$ is the Euler elastica term used to guarantee stability and smoothness.

It is important to note that the level set equations are applied for each individual connected component of the surface, thus the speed functions for the level set evolution are only available at the points of the active surface.

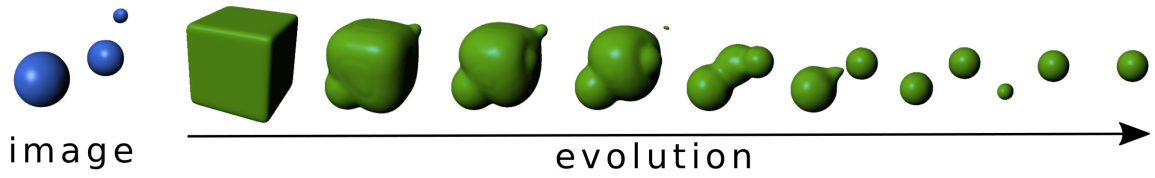
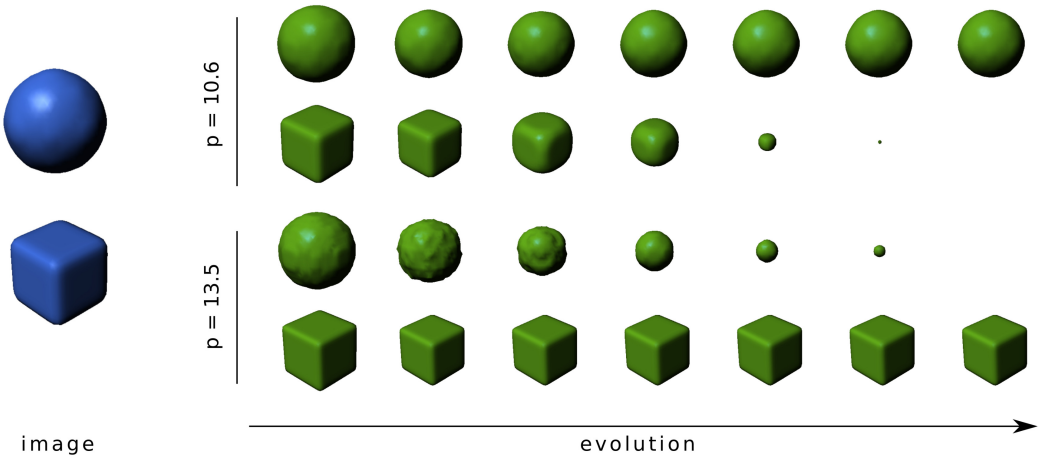
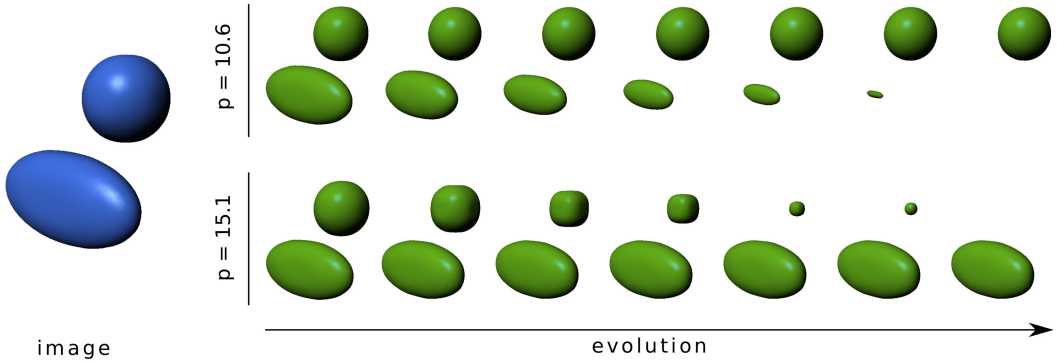


Figure 4.2: *Selective segmentation based on the volume prior. The V_0 in* (4.2) *was set to prefer the middle-sized sphere on the image.*



(a) A sphere and a cube



(b) A sphere and an ellipsoid

Figure 4.3: *Selective segmentation based on the amoeba prior. The images to be segmented are shown on the left, while the evolution of the active surfaces for distinct values of p are presented on the right. In the funcional* (5.7)*, the parameters were (a)* $\lambda = 135, \mu = 49, \eta = 1000, \theta = 50$ *(b)* $\lambda = 135, \mu = 49, \eta = 200, \theta = 50$

4.4 Euler-Lagrange formalization and level set functions

The extremal surface of a functional can be obtained by solving the corresponding Euler-Lagrange equations. In our case, these equations will have the form

$$|\mathbf{S}_u \times \mathbf{S}_v| Q\mathbf{n} = \mathbf{0}, \tag{4.8}$$

where Q is some scalar field, depending on the particular functional. The exact forms are specified in Section 4.4.1, while the methods of finding the solution is discussed in Section 4.4.2.

4.4.1 Differential equations for the functionals

The scalar field Q in the general Euler-Lagrange equation (4.8) above for the volume minimizing functional is $Q = -1$, we have $Q = -K$ for the area minimizer and $Q = \Delta I$ for the data term, where I is the image. Regarding the amoeba prior, we have

$$Q = \left[\left(\int dS\right)^{\frac{3}{2}} - p\int dV\right]\left[p - \frac{3}{2}K\left(\int dS\right)^{\frac{1}{2}}\right], \tag{4.9}$$

where $p = \frac{surface^{\frac{3}{2}}}{volume}$ is the given amoeba parameter describing the shape of the object. For the Euler elastica term, we obtain

$$Q = \frac{1}{2}K^3 - 2K_G K + \nabla \cdot \nabla K, \tag{4.10}$$

where $\nabla \cdot \nabla$ is the Laplace-Beltrami operator, which is a quantity of the tangent plane. The Euler-Lagrange equation for the Euler elastica term is calculated in Appendix A.

4.4.2 Level set formalization

To obtain the extremal surfaces minimizing the functionals, we use level set functions in combination with the gradient descent method. The Euler-Lagrange equations in Section 4.4.1 can be translated directly to level set functions by making the substitutions

$$\mathbf{S}_u \times \mathbf{S}_v \mapsto \nabla\phi, \quad \mathbf{n} \mapsto \frac{\nabla\phi}{|\nabla\phi|}. \tag{4.11}$$

The curvatures of the implicit surface can be calculated with the well known formulas

$$K \mapsto -\nabla \cdot \frac{\nabla\phi}{|\nabla\phi|}, \quad K_G \mapsto |\nabla\phi|^{-4}\begin{vmatrix}\mathbf{H}(\phi) & \nabla\phi^T \\ \nabla\phi & 0\end{vmatrix}. \tag{4.12}$$

4.5 Results

This section describes the results on synthetic and fluorescent microscopy data. We present the effects of the amoeba prior on various initial shapes and the selection capabilities of the priors.

4.5.1 Synthetic tests

Fig. 4.1 shows the effect of the amoeba prior. In these tests, the volume and amoeba priors were used. During the evolution, the objects slowly morphed until they had reached the prescribed shape parameter. To prevent the shapes from collapsing, the volume prior (4.2) was used with $k = 2$.

Fig. 4.2 shows the selective segmentation capabilities of the model on synthesized data using the volume prior (4.2) in combination with the data term. Three spheroids were present in the picture, the model was parameterized to select the one in the middle and was able to segment it out based on the prescribed volume.

Fig. 4.3 demonstrates the selective segmentation capabilities of the amoeba prior. During these tests, the functional (5.7) was used. Our model was able to distinguish between spheres, cubes and ellipsoids based on their amoeba parameter p. If the shape prior is turned on, the objects are forced towards the prescribed p parameter. If this is in conflict with the objects in the image, the volume prior (4.2) gets pushed over its inflexion point, making the object to vanish.

4.5.2 Microscopy test

The selective segmentation results on a 3D confocal fluorescent microscopy image of cells with different shapes demonstrates that our approach can be successfully used in analyzing real microscopic images. Fig. 4.4 shows volume rendering of the real images containing yeast cells. Fig. 4.5 presents the evolution of the surface while selectively segmenting out the different cell types.

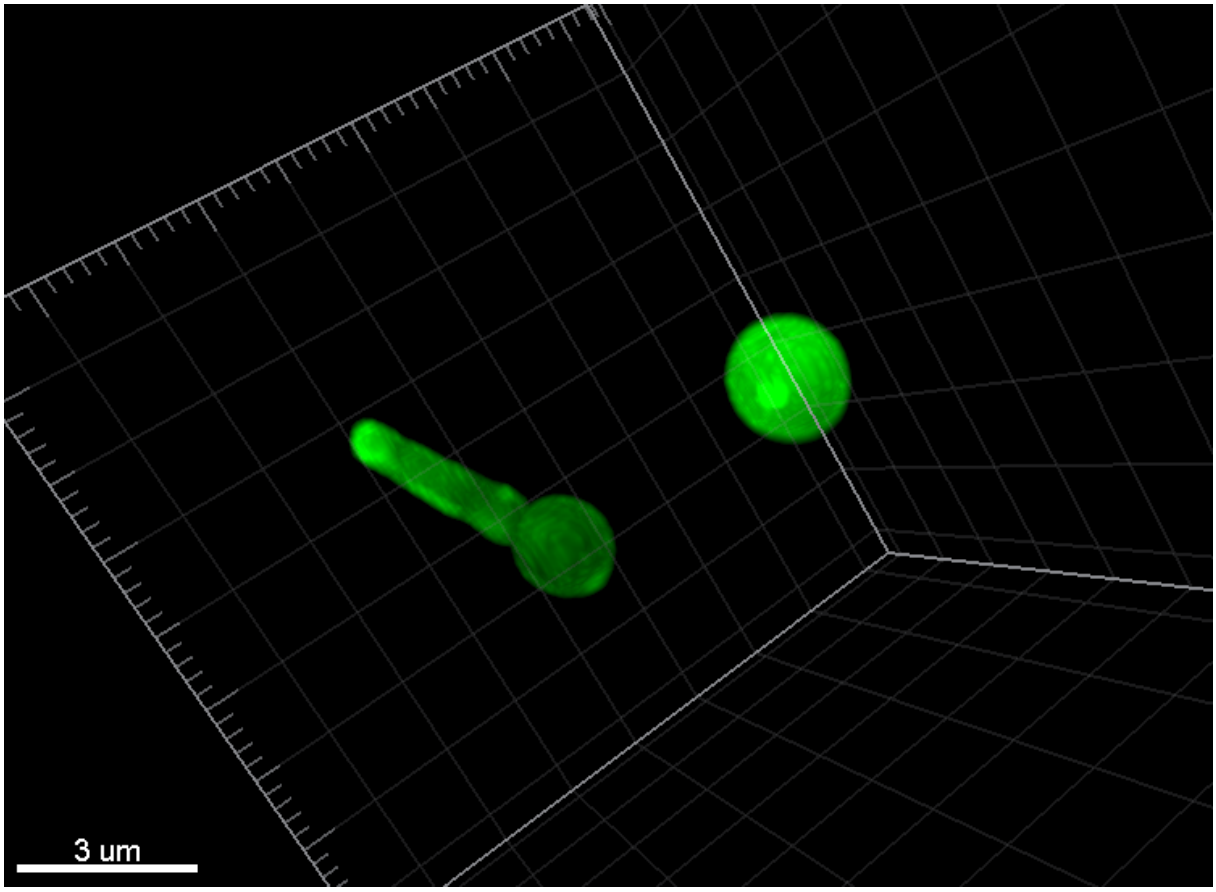
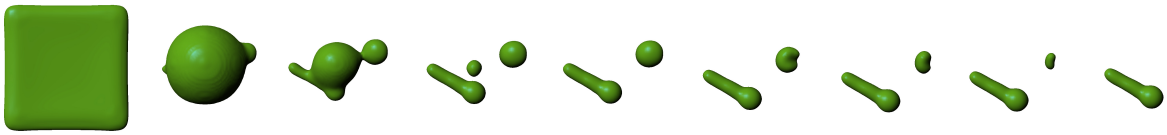


Figure 4.4: *Test image of two* Candida albicans *cells. The cell walls are stained using the Alexa Fluor™ 488 die. The left one is in* pseudohyphae *form ($p = 22.7877$), while the one in the right side is a normal yeast form ($p = 13.4637$). The goal is to recongize the two forms by segmenting them out selectively.*

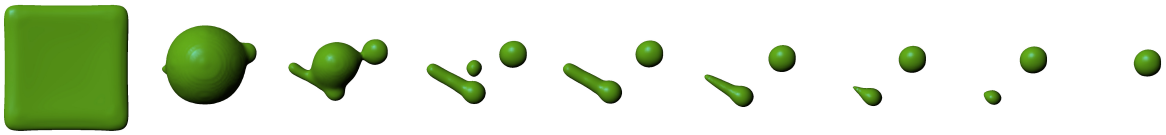
4.6 Conclusion and summary

In this chapter, using an energy minimization framework we proposed size and shape priors for selective object segmentation. Our priors were capable of segmenting various objects based on their size and shape. The derivation of the 3-dimensional Euler elastica smoothness term is found in Appending . Moreover, the corresponding Euler-Lagrange equations for the minimizing surfaces are also given.

Using active contour simulations, we showed that these priors work in practice. We showed that the volume prior can capture the object with the predefined size, while the objects with improper size are not retrieved. We also showed that using only the shape prior, we can capture objects with the proper shape parameter and our model can distinguish between cubes and spheres or spheres and ellipsoids. A



(a) Surface evolution while segmenting out the pseudohyphae form.



(b) Surface evolution while segmenting out the yeast cell in normal form.

Figure 4.5: *Selective segmentation of cells in the yeast microscopy image based on the amoeba prior. In the funcional* (5.7)*, the parameters were (a)* $\lambda = 0.005, \mu = 49, \eta = 1770, \theta = 50$ *(b)* $\lambda = 0.045, \mu = 49, \eta = 1000, \theta = 50$.

simulation is also performed to visualize the morphing of the segmentation surface shape when different target shape parameter targets are given.

Combining the volume and shape prior, we could distinguish between different forms (having sphere like or elongated shape) of a yeast cell in a real 3D fluorescent image. Our results can be also embedded in semi automatic segmentation tools to annotate 3D images that we develop in the next chapter.

Chapter 5

A semi-automatic segmentation method for multicellular fluorescent 3D images

5.1 Introduction

In chapter 4 we introduced a 3D active contour model. The model incorporates a shape prior and a size prior. It was shown that it can be used for selective object segmentation by differentiating between objects with different shapes in real fluorescent microscopy images of yeast cells in their different forms (shapes). (Figures 4.4 and 4.5) Our active contour model is implemented using the level set method to efficiently handle topological changes during the contour evolution enabling multiple object segmentation. As we discussed in chapter 2 (in the description of level set methods), one has to reinitialize the level set to signed distance function periodically. Reinitializing the contour is a costly process because one has to first extract the contour, then the minimal Eucledian distance should be computed on the grid for each point (exact reinitialization). To overcome this problem, approximate reinitialization methods are developed. Depending on the application, however, approximate reinitialization often not suitable to solve the problem properly and one has to fall back to exact distance computation. In our selective method, accuracy is crucial: among the tested reinitialization methods, only the accurate reinitialization worked properly. We observed that when approximate reinitialization is used, the shape prior simply did not have effect, that is, the shape of the contour did not converge to the proper solution defined by the shape parameter. To solve this issue, we decided to solve our PDEs in the phase field framework [86]. Then, in chapter 3 we introduced a reinitialization method that can be computed efficiently. We showed that in contrast to previous reinitialization methods, our proposed phase field reinitialization method has minimal side effect on the active model. In particular, we showed that

other reinitialization methods often contain a movement dependent on the Laplacian of the curve. When the reinitialization is executed, the curve is smoothed at the same time. This effect is undesirable in our case, because this force works against the shape prior. For example, when the shape prior defines an object with large sum curvature. After eliminating most of the curvature driven motion from the phase field functional, the selective segmentation model can be accurately and efficiently implemented. Combining these results with a simple region based data term and efficient level set implementation (local level set computation), we propose a user guided 3D image annotation software: *3D Cell Annotator* [102], similar to the annotator tool previously proposed for 3D medical images in Slicer[109]. However, there are a few major differences between Slicer and our tool. First, our method offers more control to the user since shape priors (sphericity, surface volume, surface area) can be defined before or even during the segmentation thus the surface evolution can be guided explicitly while the segmentation plugin of Slicer offers only geodesic active contours [17]. Second, our software is designed to retrieve objects from images where multiple objects are present and they can also overlap. Using the priors, for example, the user can prevent the surface evolution to fit on multiple objects by limiting its volume and/or its sphericity. There are a few other tools proposed for similar purposes, mostly relying on basic image processing primitives [23, 82]. We show that while these software offer automatic instance segmentation, after their parameters are tuned for the image, a similar accuracy can not be easily reached to our semi automatic active contour based software. Although our method needs user interaction in semi automatic mode, a similar segmentation accuracy can be achieved to fully manual drawing on 2D projections, but in significanly reduced annotation time. Our active contour model is published as an open source library that uses CUDA acceleration. A plugin is developed for the Medical Image Interaction Toolkit (MITK) [76] providing a rich 3D user interface. The MITK plugin calls the active contour model thrugh of its public API. See the ”Implementation details section” for the details.

5.2 The active contour model used

In this section, we use the results from chapter 4 for the description of the used model. See the notations there.

5.2.1 The selective functional

There are four terms included in the functional of the selective active contour model. Below we provide a concise summary. For the details, see the original paper [70].

Volume prior

The volume prior prefers objects having a certain size and it is denoted by V_0.

$$\mathcal{V}(\mathbf{S}) = \frac{1}{kV_0^k}\left(\int dV - V_0\right)^k \tag{5.1}$$

The k is an arbitrary integer. If it is 2 then the contour tries to have a volume of exactly V_0 while if it is 3, it has an inflection at V_0 therefore it prefers a volume of 0 except at V_0 where the term has no effect.

Shape prior

The shape prior penalizes the deviation of the current surface from the preferred shape p (*plasma* or *amoeba value*). The currently implemented shape prior is called *sphericity* in 3D-Cell-Annotator but essentially it is the surface/volume ratio of the surface that is calculated as $p = \frac{area^{\frac{3}{2}}}{volume}$. The plasma value is minimal for the sphere, that is $p = 3\sqrt{4\pi} \approx 10.6$. We normalize the prior to assign 1.0 to the sphere so the sphericity is given by $p - 9.6$.

The shape prior that considers the surface volume ratio is then:

$$\mathcal{S}(\mathbf{S}) = \frac{1}{2V_0^2}\left[\left(\int dS\right)^{\frac{3}{2}} - p\int dV\right]^2 \tag{5.2}$$

Smoothness term

A curvature based smoothness term is applied to prevent the instability of the surface called Euler elastica. For the derivation of the Euler elastica in 3D, see the Appendix A.

$$\mathcal{E}(\mathbf{S}) = \frac{1}{2}\int K^2 dS, \tag{5.3}$$

where the K is the sum curvature of the surface at a given point.

5.2.2 Data terms

Two different data terms are tested, but there is a wide range of possible ones to choose from. The first one is the simple edge detector:

$$\mathcal{D}_{\mathcal{E}}(\mathbf{S}) = \int \nabla I \cdot \mathbf{n} dS, \tag{5.4}$$

where the I is the image.

The second one is a region based data term. It considers the mean intensities in the rectangular prisms positioned both in the inner and the outer side of the surface. In our setting the functional maximizes the intensity difference between the inner and outer part. If this data term is used, then the algorithm has three more parameters that define the size of the region (width, height and depth).

$$\Phi(\mathbf{S},\mathbf{n}) = \frac{1}{4pqr}\left(\int_{\Re^+} I(\mathbf{p})dV - \int_{\Re^-} I(\mathbf{p})dV\right), \tag{5.5}$$

where $dV = d\xi d\zeta d\eta$, $\xi \in [-p,p]$, $\zeta \in [-q,q]$ and $\eta \in [0,r]$. $\mathbf{p}$ is in the local coordinate system, therefore $\mathbf{p} = \mathbf{S} + \xi\mathbf{e_1} + \zeta\mathbf{e_2} + \eta\mathbf{n}$ as it is visualised in fig. 5.1.

Therefore, we can use the local region as a data term in the selective model:

$$\mathcal{D}_{\mathcal{R}}(\mathbf{S}) = \int \Phi(\mathbf{S},\mathbf{n})dS. \tag{5.6}$$

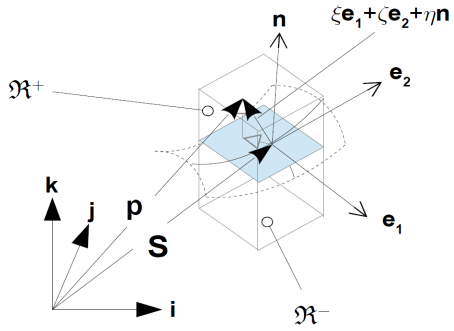


Figure 5.1: *Visualisation of the local region in 3D. The local Cartesian coordinate system of the region is centered at the surface point* $\mathbf{S}$ *while the orientation is determined by the unit normal vector of the surface* $\mathbf{n}$ *and the unit basis vectors* $\mathbf{e_1}, \mathbf{e_2}$ *of the tangential plane of the surface.*

Putting it all together: the composite functional

The composite functional therefore consists of the previously introduced terms and becomes the following:

$$\mathcal{L} = \alpha\mathcal{D} + \beta\mathcal{S} + \gamma\mathcal{V} + \delta\mathcal{E}, \tag{5.7}$$

where each term has an arbitrary real weight that can be controlled from the interface of the 3D-Cell-Annotator. The $\mathcal{D}$ can be either $\mathcal{D}_{\mathcal{E}}$ or $\mathcal{D}_{\mathcal{R}}$ depending on which data term is used.

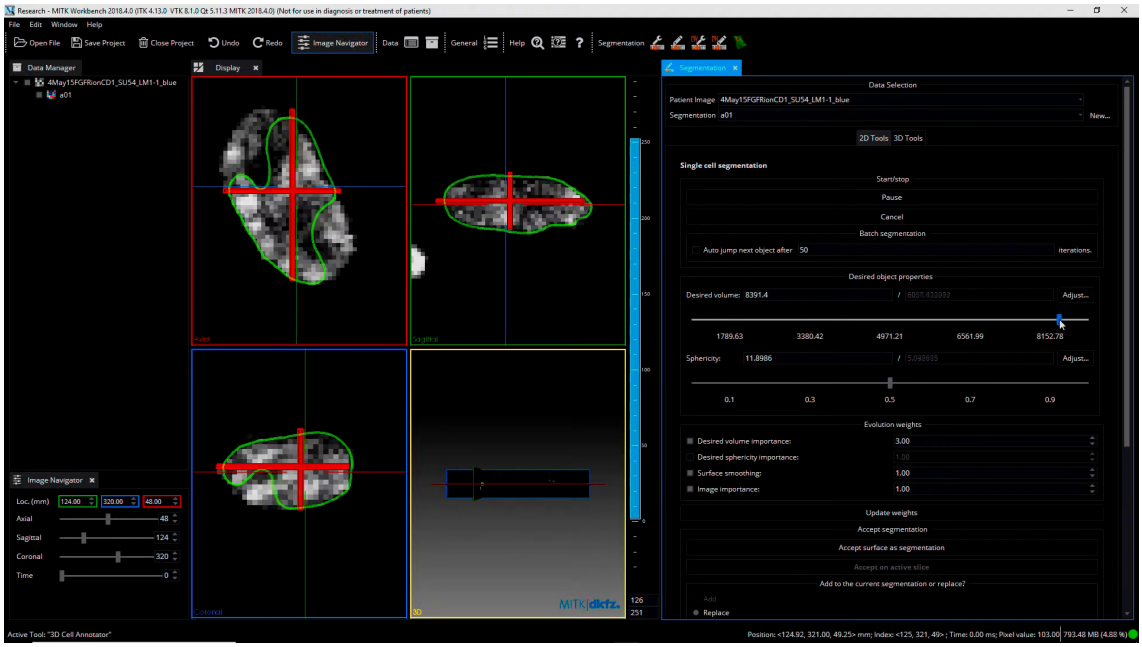


Figure 5.2: *Semi-automatic segmentation of a single-cell in a spheroid. The object under segmentation is visualized by MITK using a 3-view projection. On the left side, the user controls the surface evolution by interactively adjusting the parameters of the selective segmentation model.*

5.3 The proposed software

In this section, we briefly introduce the segmentation process and the implementation details. The detailed description of the proposed software with user manuals and tutorials are available in the supplementary information of the published paper[102].

5.3.1 Semi-automatic segmentation process

After the MITK is loaded, the user loads the image. Any format can be used that MITK otherwise recognizes. Once the 3D image is loaded, it is shown on a display splitted to four parts. In the first three parts a 2D projection from different views is shown (axial, coronal and sagittal planes). The user can navigate between the slices in each of the projected views (on the depth axis). On the fourth view, the actual segmentation is rendered as a surface in 3D. To annotate objects, the user selects a the segmentation plugin where the annotator plugin can be activated by clicking on its icon. Once the 3D-Cell-Annotator tool is activated, the control interface is appearing.

Annotating new objects

The user should first create a new label using the MITK's multilabel segmentation plugin. Then, using any of the segmentation tools, the user paints an initial contour on the 3-view projection planes. The initial object can be exactly the flat segmentation drawn by the user, or a sphere having the same diameter of the drawn initial segmentation. Once the 3D Cell Annotator plugin is activated, the segmentation is executed by clicking on the appropriate button. The surface evolution is controlled interactively by adjusting the parameters of the selective active contour model on the right side (Figure 5.2). When the user satisfied with the results, it can accept the current segmentation and the segmentation is painted to the segmentation canvas with the intensity assigned to the label the user created using the multilabel segmentation plugin.

Refining existing objects

The user can decide to refine an existing object. This can be done by first selecting the object in the multilabel segmentation plugin. Then, the algorithm is initialized using the selected (existing) segmentation. Once the user satisfied with the results, the new segmentation can be accepted by clicking on the appropriate button, and the existing segmentation is deleted for the object and the new is drawn to the canvas (or both object can be kept if the user decides to).

5.3.2 Implementation details

The 3D selective active contour model is implemented in CUDA [75] with a local level set method [95]: we first extract the contour using the marching cubes algorithm [59], then the derivatives are computes in a narrow band of the contour. The thickness of the narrow band is determined by the maximum order of the derivatives used in the active model, the values farther from the contour are not needed for moving the contour, thus these values are not computed. In the proposed software we expect that the user annotates one object at a time, thus we do not use connected component analysis.

C++ interface

The whole active contour algorithm is built as a shared library of combined CUDA device and host code, that exposes a simple API. See listing 1. The client calls the initializer function `segmentation_app_headless_init`, then the function `segmentation_app_headless_step` is called in a loop that performs one gradient descent step. The user can provide a `SimpleConfig` structure to update the active contour evolution

properties such as hyperparameters and shape priors (volume, sphericity). The actual dense level set can be requested by calling the `segmentation_app_grab_level_set` function.

MITK integration

We implemented an interface for the algorithm to the Medical Insight Toolkit (MITK [76]) to visualize the surface evolution in 3D and let the user control the parameters, load the 3D image, and export the segmentation results and open sourced the complete software. Using the C++ interface, it can be easily embedded into compatible software. [1]

Dynamic level set

MITK loads the opened 3D image into the main memory, while the active contour model runs on the GPU with limited dedicated memory on consumer machines (e.g. laptops). To solve this issue, we only copy the region of interest to the GPU memory. In particular, the bounding box of the actual contour is computed in each iteration, and the minimal padding is applied needed for the level set transition (this determines the region of the image that should be minimally extracted.). Then, we add more padding for flexibility and the resulting region of interest is copied to the GPU. If the contour grows, and the determined minimum bounding box does not fit into the previously cropped ROI, then a new crop is extracted. A similar mechanism is used when the segmentation surface volume decreases. This adaptive approach minimizes the computational time needed for one iteration of the active contour evolution.

Listing 1: The C++ interface *selective.h* for the selective active contour library.

```
typedef struct p_ObjectStat {
        float vol;
        float surf;
        float plasma;
        int gw, gh, gd;
} p_ObjectStat;

typedef struct p_int3 {
        int x;
        int y;
        int z;
} p_int3;
```

[1] Source code: `https://github.com/etasnadi/3DCellAnnotator`

```
/**
 * Initializes the algorithm.
 * image: an unsigned integer image, each pixel is an eight bit gray value.
 * segmentation: an unsigned integer image, each pixel is a 16-bit label id.
 It is the level set initializer depending on the configuration.
 * labelId: selects the object provided in the previous label set image.
 * pixelSize: unused
 * imageSize: the dimensions of the image and the segmentation
 * a_conf: the configuration
 */
EXPORT_SHARED int segmentation_app_headless_init(
    const void* image, const void* segmentation,
    int labelId, int pixelSize, p_int3 imageSize,
    SimpleConfig a_conf);

/**
 * Performs a simple evolution step. The configuration will be updated with
 the one passed in the argument.
 * Returns an object statistics.
 */
EXPORT_SHARED p_ObjectStat segmentation_app_headless_step(
    SimpleConfig conf);

/*
 * Returns the current level set as a float array. The level set size will be
 placed in the output variable gsize, and the translation will be put to the
 trans variable.
 * The algorithm only maintains a level set where there is an object. The
 frame will be extracted and may be updated during the segmentation. The
 gsize variable tells the
 * client the frame size, while the trans variable holds the translation
 to the image origin to the frame origin.
 */
EXPORT_SHARED float* segmentation_app_grab_level_set(
    p_int3& gsize, p_int3& trans);

/*
 * Releases the resources allocated.
 */
```

```
EXPORT_SHARED int segmentation_app_headless_cleanup();
```

5.4 Results

5.4.1 Evaluation metric

We measure the closeness of two objects (X and X) using the Jaccard similarity by the total number of pixels minus the number of overlapping pixels over the number of overlapping pixels:

$$\text{Jaccard-similarity}(X, Y) = \frac{|X \cap Y|}{|X| + |Y| - |X \cap Y|}. \tag{5.8}$$

For multi-object images, the best matched objects are selected, then the Jaccard-similarity of the matches are computed.

5.4.2 Datasets

We used three publicly available datasets for testing:

- A confocal dataset of 77 3D images, each contains exactly one single cell [84].
- A LSFM spheroid that contains 52 cells [32].
- A confocal mouse embryo image that contains 56 cells [89].

5.4.3 Competing methods

We compare the segmentation accuracy obtained with 3D-Cell-Annotator to competing methods using basic image processing primitives: MINS [60], Pagita [33], XPIWIT [6] and OpenSegSPIM [32]. See our review [82] for a detailed description of these tools.

5.4.4 Single-cell dataset

As the ground truth is not available for the dataset, we asked three experts to annotate the data manually. The manual annotations are then averaged using publicly available MATLAB tool [83]. We semi-automatically segmented each cell using 3D-Cell-Annotator. The average Jaccard similarity between the expert annotations are 0.76. The Jaccard similarity of the segmentations obtained using 3D-Cell-Annotator to the averaged annotations is 0.77, while the annotation time reduced from 8 hours (fully manual annotation) to 4 hours (semi-automatic segmentation using 3D-Cell-Annotator).

5.4.5 Multicellular segmentation

On the spheroid dataset we have a ground truth annotation. We asked two experts to reannotate the images. The average of the experts reached an accuracy of 0.658 (in terms of Jaccard similarity), and the distance of the annotations obtained by 3D-CellAnnotator from the experts are 0.689 and 0.677. We tested four competing automatic segmentation tools for which we tuned the parameters. Although they return the segmentations almost instantly, we could not reach accuracies larger than 0.563 and 0.449. On the embryo dataset, the accuracy between the average of two experts and the ground truth is 0.80. The Jaccard similarity of the segmentations obtained by our software to the experts are 0.802 and 0.799 while the annotation time is reduced from 6 to 3 hours compared to fully manual annotation.

Part II

Deep learning based methods

Chapter 6

Machine learning based techniques

Here, we describe the limitations of active contour methods and introduce machine learning and explain how we can use learning based techniques for single cell segmentation.

6.1 Limitations of active contours

In part I an active contour method is introduced to segment objects based on prior information: not only the shape priors can be considered here as priors, but even the whole active contour framework. In the active contour framework, for example, we assume that the objects can be represented as a smooth, closed curve (or a surface in 3D). More importantly, we always used a data term (for example the one labeled with 2) that connects the image and the contour itself. The data term is designed to return large values when the contour is in a non-desired location (for example located in the background) and minimal value at the object boundary. Edge-based data terms assume that the magnitude of the gradient ($|\nabla I|$) of the image I is greater at the boundary of the object and lower at homogeneous regions such as locations in the background or in the inner region of the objects. This assumption may hold for simpler images, however, on less trivial cases (e.g. images where objects have blurry boundary), the data term should be conditioned on a bunch of pixels. This observation led to the development of region based active contours [20, 70, 87]. The simplest region based data terms consider an inner and an outer region for each contour location, then they return low values when the differences between the mean or sum intensities in the regions are larger. A region based data term, however, in theory, can compute an arbitrary function of the pixel intensities not limited only to computing the sums or means. More recent models thus consider higher level region intensity statistics to find the contour boundaries [12, 54]. Even if local statistics are used for localizing object boundaries, we still rely on prior information (the intensity difference or the chosen statistic). In part II: we loosen these assumptions on prior

information and introduce machine learning based techniques for nuclei and single cell segmentation to learn the model directly from the data. That is, the objects are not modeled by using a prior statistic along the curve for separating the inner and outer region for an object but learned directly from the data.

6.2 Supervised learning

Assume that we have a dataset $\mathcal{D} = \{(\mathbf{x}_i, y_i)\}_{i=1}^N$ of consisting of N samples. Each sample has M features $\mathbf{x}_i \in \mathcal{X}$ and an observed label $y_i \in \mathcal{Y}$. We have given a hypothesis space $\mathcal{H}$ usually coupled with the learning algorithm, and our task is to learn the *best* hypothesis explaining the data $h : \mathcal{X} \to \mathcal{Y}$ according to some error/loss function $\ell : \mathcal{Y} \times \mathcal{Y} \to \mathbb{R}$ to be minimized:

$$h^* = \underset{h \in \mathcal{H}}{\operatorname{argmin}} \sum_{i=1}^{N} \ell(h(\mathbf{x_i}), y_i) \tag{6.1}$$

Depending on the particular machine learning method, $\mathcal{H}$ can be the space of linear functions $y = \mathbf{x}^T\mathbf{w} + b$ (when $\mathbf{x}_i$ and y_i are continuous) or all of the possible decision trees (when $\mathbf{x}_i$ and y_i are discrete). When training supervised models, one should select a number of samples where the labels are observed, and search for the best hypothesis in the space of all hypotheses $\mathcal{H}$. The goal however is to look for a hypothesis that generalize well to data where the labels are not observed. One can separate $\mathcal{D}$ into a training and a test set to mimic this process. The hypothesis is fit on the training set and the accuracy is measured on the test. A better choice is to use cross validation whenever possible: using cross-validation we are no longer dependent on the selection of the training set, thus we can minimize the variance among the performances of each fold making the found hypothesis less dependent on a particular selection of the training set in each fold leading to more stable models. In contrast to the active contours, the most important prior knowledge is the hypothesis space.

6.3 Supervised image segmentation

While active contours can be naturally applied to image segmentation, supervised learning is a general technique that can be exploited in image segmentation several ways. We introduce the most popular of such techniques in microscopy image analysis.

6.3.1 Basic machine learning

In the active contour methods, the hypothesis space was the set of all possible closed continuous planar curves in the image domain. When using classical supervised methods, one can classify each of the pixels in the image and the class either can be 'background' or 'foreground'. This leads to a segmentation map that is called *semantic segmentation*. Semantic segmentation, however, does not have the expressive power to represent regions like active contours naturally do, thus a subsequent processing step is needed to combine the image pixels into regions. A myriad of methods are applied for this problem e.g. connected component analysis, watershed segmentation, object splitting, graph-cut methods, and correlative clustering among others. The resulting technique is often called *instance segmentation* to differentiate between simple pixel labeling and single object retrieval. One can use a linear model to classify pixels for separating the foreground and background e.g. logistic regression. However, it was shown, that decision trees are superior among other supervised learning algorithms for tasks in systems biology including microscopy image analysis [30]. A decision tree is usually regularized by minimizing its size to prevent overfitting on the training data causing the limited expressive power of the trees. Random decision forest classifiers shown to improve the training and test accuracy at the same time by training multiple decision trees, then applying a voting mechanism during prediction [37]. A seminal method in microscopy image segmentation uses a random forest classifier to classify pixels based on intensity, edge and texture based features. The resulting semantic segmentation map then can be further processed to achieve instance segmentation using the above mentioned methods. One major limitation of these models is that we still have to design proper feature extraction for each task.

6.3.2 Deep learning

Linear models can not represent complex decision surfaces. Multilayer perceptrons (or artificial neural networks) are multi-layered linear models consisting several non linear models (units), where the output of a particular layer is given to the input of the subsequent layer, except for the input layer and the output layer. The non-linearity is achieved by applying a threshold function (or activation function) on the output of the unit e.g. $y = max(\mathbf{x^T w} + b, 0)$ (known as liner rectifier) or a function from the sigmoid family. The simplest model consinsts of an input an output and a hidden layer placed between the two. The network is fully connected that means, each unit in a layer receives input from all units in the previous layer and not from units in other layers. It was shown that even with one hidden layer, arbitrary decision surfaces can be modelled. The network capacity can be increased by adding more unit in each layer (increasing the width) or increasing the number of hidden layers (increasing the depth). The network is fully differentiable and it can be trained us-

ing backpropagation. The features are first extracted, and the values are fed to the inputs to the neural network. However, this two step learning has a major drawback: determining the exact features for each task needs expert knowledge. It is not always exactly known that what combination of features lead to the best generalizing models. Since the capacity of artificial neural networks can be increased easily (by adding more units), one can show the image pixels directly to the input of the network. This one-step training eliminates the need of manual feature design because in this way the machine learning model not only conditioned on the joint distribution of the features and the labels but directly on the input data. Thus, the network can 'learn' the functions previously identified by experts, or may learn novel features. End-to-end learning with multilayer perceptrons still have at least one drawback: the number of parameters can be very high because of the fully connected nature of the network topology that causes difficulties in training but also reduces the generalization ability of the model. One can reduce the number of units in the network by employing *shared weights*: one unit is used to process multiple parts of the image presented to the network. This approach leads to a decreased number of units thus fewer parameters and a more regularized network. The connections, however, can be further decreased by connecting the units only to proximal inputs at the previous layer. With these two improvements, each unit can be thought of convolutions, thus this family of networks are called convolutional neural networks (CNNs). CNN architecture is invented in 1980 by Fukushima et al. and LeCun used backpropagation to train them in 1989 [29, 51]. With the advent of General Purpose GPU (GPGPU) computing CNNs are trained on GPUs achieving 10x-50x performance compared to CPU training. Ciresan et al. proposed DanNet in 2011 and won several image classification benchmarks by a large margin [22]. Not much later, an equivalent model is trained by Krizhevsky et al. and the authors showed superior performance on natural image classification on ImageNet (outperformed the second best submission by more than 10 percents in error rate) [25, 48]. It is known that deeper networks have better generalization ability, however gradient based optimization is more challenging of such deep networks because of the chain rule of differentiation. An effective solution for this problem is the selective connection of the layers using so called skip connections [98]. The authors shower that they can train neural networks with hunderds of layers by using the gradients from the top layers thrugh the skip connections in the bottom of the network, and they popularized this invention as Highway networks. Later, the special case of the highway networks is proposed as ResNets that won the ImageNet 2015 challenge in natural image classification [35].

6.3.3 Convolutional neural networks for image segmentation

We saw that CNNs improve the classification accuracy on natural image benchmarks. The input of the network is an image and an output is a vector that encodes the probability distribution of the labels conditioned on the input by the parameters of the CNN. In such architectures, most of the layers are convolutional layers outputting dense feature maps replacing the classical feature extraction pipelines. The feature maps are extracted in the last convolutional layer, then several fully connected layers are used for determining the class probability distribution [96]. For segmentation, the last convolutional feature map can be used to supervise the values on pixel level, thus the network transforms the input image to semantic segmentation directly with convolutional and downsampling layers. The main issue here is that the last spatial size of the last convolutional layer is often a small fraction of the input signal (e.g. 1/8 or less). This is a problem, because the network can only output coarse segmentation maps. The solution is to use skip connections but not only for the mere purpose of eliminating the vanishing gradients issue but to use fine-grained features from layers closer to the input when predicting the class distribution of a pixel in the output segmentation map [58]. In other words, the skip connections are essential, because subsequent feature pooling layers downsample the input feature map to extract compact feature representation from the input needed for constructing abstract clues at the cost of losing fine features from higher levels. This may led to coarse semantic segmentations, thus the responsibility of skip connections is to retain as much information as possible and then fuse abstract knowledge with fine pixel level feature representation for constructing each element of the semantic segmentation map. An improvement of the latter method is the DeepLab network, that uses atrous convolution to reduce the downsampling ratio of the network thus predicting more fine-grained segmentation maps. The method uses fully connected conditional random fields to further refine the quality of the output [21]. Another improvement learns the pooling layers as well [4]. A similar method uses a symmetric network consisting of an encoder (downsampling after convolutions) and a decoder (upsampling after convolutions) with skip connections called U-Net [88]. It showed superior performance on several biological image analysis benchmarks in instance segmentation. The network assigns the label foreground or background for each pixel. During supervision, in the target segmentation map, the instances are separated by background pixels and a higher weight is associated for them in the loss forcing the network to concentrate on classifying the separating pixels properly. During prediction a connected component analysis is performed on the resulting segmentation to extract the instances. These networks produce pixel level outputs, then combine the predicted pixels to regions as a post processing step. The downside of this two-step approach is that the detection part cannot be efficiently done without breaking the gradient flow.

Another class of methods e.g. region based CNNs (R-CNNs [31]) consist of several subnetworks each having their own responsibility in the instance segmentation process. The first network is responsible to extract dense features, then object bounding boxes are fitted on the feature map using a shallow network in a sliding-window fashion followed by the non-maximum suppression, and a subsequent semantic segmentation network is used for semantic segmentation in each fitted bounding box. On the Data Science Bowl 2018 nuclei segmentation challenge, the top performing entry used an ensemble of U-Nets and several other teams focused on smart data augmentation strategies and used simple Mask R-CNN (a variant of R-CNN) [1, 34]. To see how such models perform on the nuclei segmentation task, see a recent review [38]. In the following chapter, we introduce data augmentation strategies to improve the test accuracy on the multi-modal DSB 2018 [14] dataset by generating augmented images automatically, then we learn the whole augmentation process by a combination of networks.

Chapter 7

Automatic domain adaptation using image style transfer

7.1 Introduction

In this chapter, we propose a data-driven image augmentation protocol called *nucleAIzer* [39] for nuclei segmentation that uses the test set in order to augment the initial training set. Our goal is to enrich the starting training set with images similar to the test set bridging the distribution gap, hence we classify our method as domain adaptation. We expect that if a model is trained such way, it will perform better on the task. Our method is tested on the dataset published for the Data Science Bowl 2018 nuclei segmentation competition (DSB 2018) [14]. The dataset consists of two subsets, the stage1 and the stage2. The first stage of the competition is shared with the community to develop the model and contains 670 images and also the corresponding segmentation mask for each image in the training set, while it has 65 test samples. For the second stage, the training set is the same, but there are different 65 images for testing. The dataset consists of microscopy images with several different modalities. In the training set(test set) 80.6% (67.9%) of the images are fluorescent stained images, 0.6% (4.7%) are hematoxylin and eosin (H&E) stained tissue (nuclei instances stained in the tissue are purple, while the bacgkround has a pink shade), 15.5% (15.1%) has purple and white shade (tissue), 0.9% (11.3%) contains fluroescent images with larger than usual nuclei and 2.4% (0.9%) are brightfield microscopy images. Our method potentially synthesize samples that are completely missing from the training set by a form of self learning using weak labels.

7.2 The proposed method

Our proposed method consists of several processing steps. We also use external data to reach a baseline accuracy, and then the main dataset is used to fine tune the model. See figure 7.1 and pseudocode 1 for the details. First, the instances M^{weak} are detected in the elements of the test set using an instance segmentation network [34] (m_{preseg}) on the external training data S^0. Then, the images are resized to have the instances uniform expected size s. The next step is the synthesis of artificial images. We do it for each different modality in the test set S^{test}: the most distinctive styles in the dataset are identified by first clustering the microscopy images in S^{test}. Next, an image-to-image translation [41] model is trained on the elements of the training and test set (using the weak instances generated by m_{preseg}) that maps the masks into the corresponding microscopy images. Then, we measure the parameters of the instance masks in each style (e.g. expected number of instances, shape features and distribution of the object locations and sizes), and we synthesize instance masks using the computed parameters by copying objects from the training set masks combined with an external nucleus database M^0 to empty canvases. Using the synthesized instance masks and the learned image-to-image translation model, we synthesize the corresponding microscopy image for each synthesized mask. The synthesized samples are then combined with the starting training set, then the weak instance segmentation network m_{preseg} is fine tuned on the augmented enriched training set.

The method starts by training m_{preseg} on the external dataset S^0.

7.2.1 Sample uniformization

In the first step, we uniformize samples in the training and test set. For doing that, one should know the approximate object sizes, compute the expected object size, and then resize the images and the masks to have the expected object size uniform across the dataset. This preprocessing step may increase the accuracy of the model and it is not uncommon in the nuclei segmentation community, at least a different model uses similar resize strategy. [99]. This is trivial for the training set, but obviously not for the test set where the ground truth instances are not observed. Although the expected object sizes in biological image analysis is known in special cases, for example when the same type of tissue/cell is imaged with known imaging pipeline parameters (e.g. microscopy magnification level), the purpose of the DSB 2018 challenge is to develop a foundational model that can be generally applied to different images [14]. Hence, in the general case, the expected object size is not known for the test set. Therefore we opt to determine the object sizes in S^{test} by using m_{preseg}.

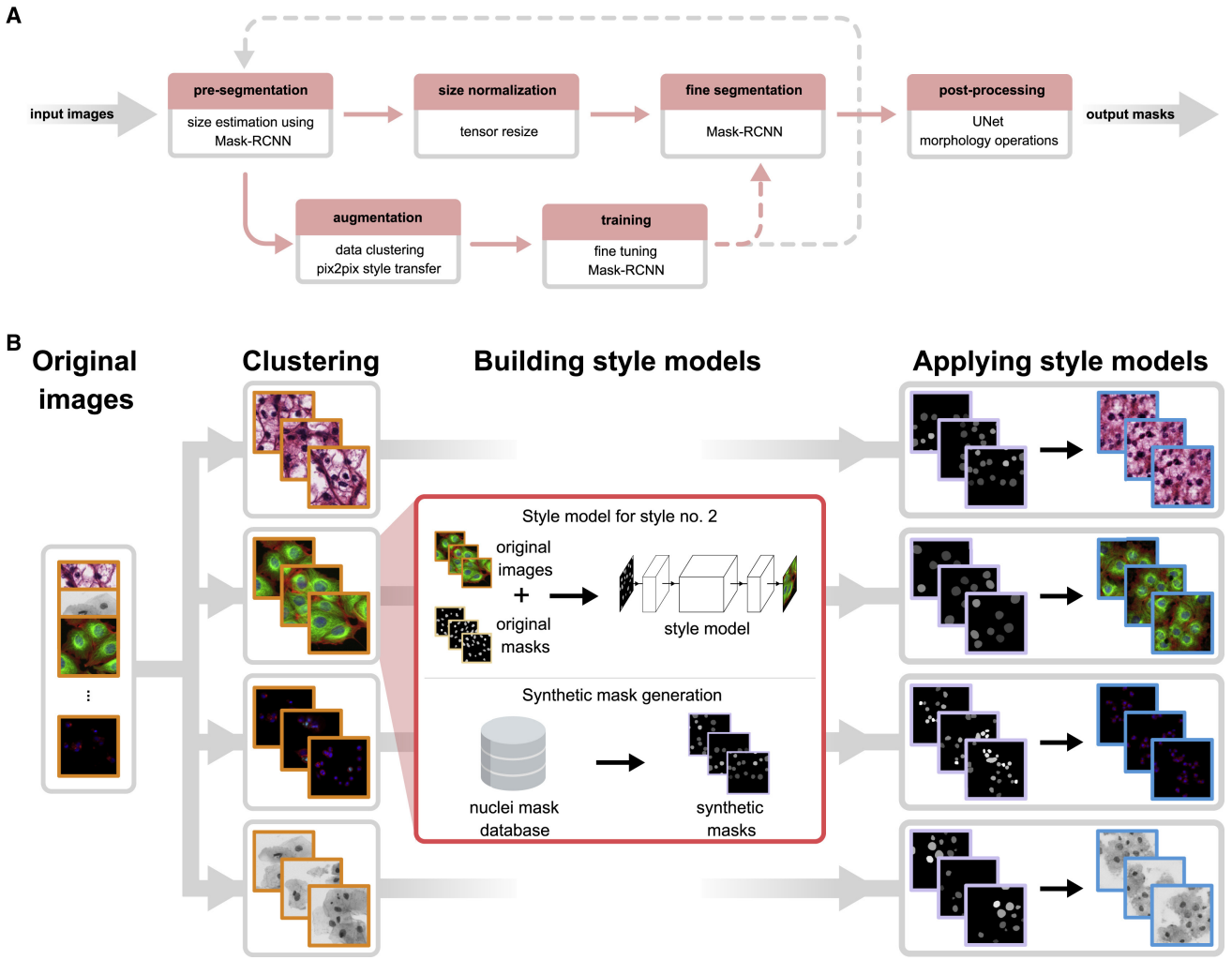


Figure 7.1: *Datasets: top: salivary gland, bottom: fallopian tube. The datasets are from the article [74]. The right side shows the input image crops, the corresponding labeled masks, and the structure representation of the masks first proposed in [99].*

7.2.2 Style identification

To enrich the training set with relevant examples, we have to identify the styles in the test set, that is, each style represents a distinct image modality. The styles are first manually identified in the training set S^{train}, then the features are extracted using CellProfiler [15, 50]. A shallow neural network is then trained to model the equivalence relation across the styles. That is, the network is trained to predict 1 for features extracted from similar images in the training set, and 0 otherwise. During prediction, we use this network for the elements of the test set and construct a similarity matrix. The clusters are then identified by applying adaptive *k*-means on the similarity matrix (k=134 in our experiment for DSB 2018 stage 2 test set).

7.2.3 Image-to-image translation task

After we identified the distinct styles $S^{test,clus}$ in the test set, we learn an image-to-image translation (pix2pix [41]) model $m_{pix2pix}$ that is trained to transform masks into the microscopy images. Since we have no ground truth for the test set, we use the weak instances M^{weak} for the source of the pix2pix transformation.

7.2.4 Synthesizing novel samples from the test distribution

For each of the 134 identified clusters in the DSB 2018 test set, we have a style model that is trained to transform the weak labels to the microscopy images. For synthesizing novel samples, we still have to acquire the masks that will be fed to the image-to-image translation network. To synthesize the masks, we use two methods. The first one simply draws objects from an external nuclei database M^0 with uniform random positions and sizes (having the expected size s). The second one is a parametric cell population simulation tool developed for simulating cell populations (and for some cases also the textures) [53]. Using the latter tool one can define simple parameters for the masks including number of objects, mean object size and the clustering coefficient (that tries to capture the local groupings of objects often observed in cell cultures). These parameters are computed style-wise on the test set. We used to generate around 50% of the masks using the first method and the remaining with the latter (and we get $l = 20$ synthesized samples for each style). Then, the image-to-image translation network is used to synthesize the corresponding microscopy images by feeding the synthetic masks into $m_{pix2pix}$ in inference mode.

7.2.5 Training

We used a top-down region based convolutional neural network (Mask R-CNN [34]) for the segmentation task. The presegmentation m_{preseg} model is trained on external

data S^0, and we fine-tuned the model on the DSB 2018 training set S^{train} enriched by the synthesized samples generated by our method. The combined dataset is further augmented to reduce overfitting thus increasing the test accuracy. The algorithm is summarized in 1

ALGORITHM 1
The pseudocode of the *nucleAIzer* method. S^0 is an external annotated dataset, S^{train}, S^{test} is the actual dataset (DSB 2018 [14]), M^0 is an external nucleus database, s is the expected nucleus size after size uniformization, l is the number of images synthesized for each identified modality (style).

procedure NUCLEAIZER(S^0, S^{train}, S^{test}, M^0, s, l)
  $m_{preseg} \leftarrow$ MASK R-CNN-TRAIN(S^0)
  $M^{weak} \leftarrow$ MASK R-CNN-INFERENCE(S^{test})
  $S^{train} \leftarrow$ RESIZE-IMAGES(S^{train}, s)
  $S^{test} \leftarrow$ RESIZE-IMAGES(S^{test}, M^{weak}, s)
  $m_{clus} \leftarrow$ CLUSTER NETWORK-TRAIN(S^{train})
  $S^{test,clus} \leftarrow$ K-MEANS(m_{clus}, S^{test})
  for $c = 1$ to $|S^{test,clus}|$ **do**
    $m_{pix2pix,c} \leftarrow$ PIX2PIX-TRAIN($S_c^{test,clus}, M^{weak}$)
    $M_c^{synth,clus} \leftarrow$ SYNTHESIZE-MASKS($M_c^{test,clus}, M^{weak}, l$)
    $M_c^{synth,clus} \leftarrow M_c^{synth,clus} \cup$ SIMCEP($M_c^{test,clus}, l$)
    $I_c^{synth,clus} \leftarrow$ PIX2PIX-INFERENCE($M_c^{synth,clus}$)
    $S \leftarrow S \cup (I_c^{synth,clus}, M_c^{synth,clus})$
  end for
  $S_{aug} \leftarrow$ AUGMENT(S)
  $m_{nucleAIzer} \leftarrow$ MASK R-CNN-TRAIN(S_{aug}) **return** $m_{nucleAIzer}$
end procedure

7.3 Results

7.3.1 Evaluation metric

We evaluate the segmentation accuracy using the DSB 2018 score. The same metric used in the competition. To compute the metric, one should match each predicted object to each ground truth object:

$$\text{IoU}(x, y) = \frac{x \cap y}{x \cup y} = \frac{x \cap y}{|x| + |y| - |x \cap y|}. \tag{7.1}$$

Then, the optimal assignment is computed on the resulting similarity matrix using the Hungarian method [49, 92]. The matching is then discretized by thresholding the

IoU scores on threshold levels $T = 0.5, ..., 0.95$ with step size 0.05. On each threshold level, we compute the DSB score:

$$\text{DSB-score}(t) = \frac{\text{TP}(t)}{\text{TP}(t) + \text{FP}(t) + \text{FN}(t)}, \tag{7.2}$$

where $\text{TP}(t)$ is the number of true positives with IoU score at least t, $\text{FP}(t)$ is the number of false positives, and $\text{FN}(t)$ is the number of objects missed at threshold level t (assuming that $\text{TP}(t) + \text{FP}(t) + \text{FN}(t) > 0$), otherwise the DSB-score(t) is 0.

The final DSB score is the average of the DSB scores at each threshold level, that is:

$$\text{DSB-Score}(T) = \sum_{t=1}^{|T|} \frac{\text{DSB-Score}(T_t)}{|T|}. \tag{7.3}$$

7.3.2 Baselines

We compared our method to a six other methods. The *Ilastik* pixel classification method [7] uses a random forest classifier to classify pixels based on feature information extracted for each pixel in the image. The resulting classification map is then thresholded and the resulting regions are found by computing the connected components. The resulting objects are split by analyzing the contour geometry if necessary.

Unet4nuclei is a U-Net [88] based nuclei segmentation method described in [14].

GVF (gradient vector flow [107]) is an active contour method with a preprocessed edge map. The GVF algorithm smoothes the gradient information to the regions far from the objects, thus the contour can move towards the object even in background (homogeneous) regions where the data term is 0. In that case, without the GVF, the only force that could theoretically move the contour would be the smoothing term.

CP is the CellProfiler baseline open sourced by the DSB 2018 organizers. It consits of different pipelines for each image type. The organizers claim that no single pipleine is likely to exist that can process all the modalities optimally. However, deep learning models are expected to work on all image types [14].

7.4 Results

The results are presented in table 7.1. If we compare the presegmentation model trained on external data and the Mask R-CNN row, that is the presegmentation model fine tuned on the target dataset, the initial 0.468 (0.438) improves to 0.542 (0.621) on the stage 1test (stage2 test). When we also use the synthetic data generated by our pipeline, we can further improve the accuracy to 0.585 (0.633) on stage 1 test (stage 2 test).

	fluo	hist	stage1 test	stage2 test
presegmentation (m_{preseg})	0.654	0.351	0.468	0.438
CP	0.599	0.255	0.333	0.528
Unet4nuclei	0.653	0.052	0.138	-
GVF	0.519	0.039	0.258	0.168
Mask R-CNN	0.676	0.381	0.542	0.621
Mask R-CNN + style learning	0.626	0.421	0.585	0.633

Table 7.1: *Segmentation resutls on different subsets with different methods. The subset fluo means that the model is evaluated only on the fluorescent subset of the dataset, while hist is the tissue part of the dataset.*

7.5 Summary

We first trained a model on external data that can already achieve reasonably good DSB score when fine tuned on the actual dataset. We then developed a method that incoroporates the unlabeled samples from the test into the training by generating weak labels for them using the presegmentation model. We showed that our strategy is effective on the DSB 2018 stage1 and stage 2 since it can improve the accuracy on both subsets.

Chapter 8

Structure preserving adversarial generation of labeled training samples for single cell segmentation

8.1 Introduction

Data augmentation is one of the simplest ways to improve the generalization capability of convolutional neural networks. The motivation behind data augmentation is the fact that using the appropriate transformations one can generate artificial elements from the original dataset that are feasible to boost the networks' performance. Such common transformations used in biological image analysis are simple affine transformations like rotation, translation, scaling and nonlinear transformations, for example elastic deformations that equally affect both the input and target image. Other transformations like addition of (Gaussian) noise or intensity transformations affect the input image only and leave the segmentation untouched. In contrast to these transformations, our approach uses generative adversarial networks and image-to-image translation to synthesize novel samples that mimic the original dataset therefore it can be considered a fully data-driven approach. First, a state of the art GAN is used (StyleGAN2-ada in our case [43]) to learn the distribution and shape of the instances in the mask images of the training set by applying a general encoding technique to encode the masks in order to be able to be learnt by the network. Then, synthetic masks are generated using the learnt GAN model. In parallel to the GAN training, an image-to-image translation task is solved using the pix2pix method [41] to learn the translation from the target masks to the microscopy images in the training dataset. Then, the pix2pix models are used to construct the corresponding synthetic microscopy image for each synthesized mask image. The synthesized training samples can be used for initial pre-training or can be concatenated to the original dataset and the network can be trained on the combined dataset. In

the latter case, the network will be optimized not only to the original dataset but also to the synthetic images that may be suboptimal in certain cases. Therefore, we train the instance segmentation network on the synthetic images first, to initialize the weights, then the original dataset is used to fine-tune the pretrained network. We also tested the performance of our method when training with only a subset of the annotated images. In this case, we found that merging the training set and the synthesized samples substantially improves the segmentation accuracy. By combining our method with traditional augmentation techniques (intensity transforms combined with nonlinear deformation), we found that we can achieve slightly higher instance segmentation accuracy compared to the case when only augmentations are used in the pre-training then fine-tuning policy.

8.2 Related work

Our proposed method aims to solve two subproblems: (1) the implicit generation of cell instance masks without any expert knowledge by learning the underlying distribution of the training set using a traditional GAN and (2) synthesizing the corresponding microscopy images to the generated masks which in turn shown to boost generalization capability when training instance segmentation networks. Simulating cell populations. For the first task, there are a handful number of methods proposed: the SIMCEP method [53] aims to generate realistically looking cell populations in two-steps, in which the first step the instances are modeled using parametric curves with small displacement in each time-step sampled from a uniform distribution. Then, the generated objects are placed on a canvas with random coordinates sampled from a parametric distribution. Another interesting tool is the Cytopacq [101, 106] method that aims to simulate the whole imaging pipeline. The approach is also capable of generating 3D digital 'phantoms' of HL-60 cells (among others) by deforming a sphere or ellipsoid using fast level set methods with random noise. Although these approaches can model simple cell populations, the position of the objects is still drawn from simple parametric distributions. In contrast, our goal is to learn virtually any distribution of the objects along with their shapes and relative positions from the training dataset and generate new cell populations from the same distribution. This is especially useful when the layout of the objects on the masks follows a specific pattern that can not be easily modeled using simple parametric distributions.

Exploiting GANs for augmentation. Several methods exist in the literature that successfully exploited GANs to improve the generalization capability of convolutional neural networks. Most of the methods aim to solve segmentation or classification tasks for biomedical images. Applying GANs for the segmentation of medical images may be more straightforward as usually only one or a few well defined target objects

may be segmented with approximate locations known prior while in biological image analysis, usually an unknown number of instances shall be recognized from one or multiple classes. In medical image analysis, a common approach is to simply merge the input image and the corresponding segmentation of each sample in the training set, then learn the joint distribution using a generative model. The synthetic images and their corresponding masks then can be synthesized using the learned model [10, 18]. The segmentation mask is merged with the image channel, then feeds the resulting images to a progressive growing GAN (PG-GAN). The method works on CT and MRI datasets. Another work aims to exploit unlabelled samples from an MRI dataset to estimate the parameters of elastic deformations and intensity transformations in order to adapt the training set to the test set using conditional GANs [18]. It has been shown that meaningful representation can be learned from unlabelled samples of microscopy images by feeding all of the available images into a GAN, then copying the discriminator weights into the encoder part of a U-Net that is then fine-tuned on the annotated dataset [64]. A similar work uses CycleGAN to adapt CT images from public annotated datasets usually containing contrast images to their non-contrast version to reduce the domain gap between the public datasets and the practically as important non-contrast cases (when injecting the contrast agent into the patient is not feasible) [90]. The segmentation network is then trained on the original non-contrast images and the synthetic non-contrast images originally coming from the public contrast dataset are used as augmentation. The authors claim that this method dramatically improves on the segmentation accuracy. The effectiveness of GAN based augmentation is shown in classification tasks as well. X-ray images can be used to synthesize images with the desired pathologies in order to generate samples for less abundant classes [100]. DCGANs and conditional GANs can be used to improve classification of lesions found in liver CT slices. In the basic GAN settings, one distribution is learned for each class, while in the conditional GAN setting, a class-conditional distribution is learned from the whole dataset [27, 28]. Using GANs to generate chest X-ray images to improve the accuracy of cardiovascular abnormality detection is also shown to be effective [62]. Another work proposes a nuclei segmentation model for segmentation of histopathology images where both the synthetic data generation and the segmentation is trained adversarially, although they randomly sample nuclei instances and simply draw them on the canvas randomly with respect to the expected shape parameters for the particular organ. Then, an unpaired image-to-image translation model is trained (CycleGAN) to learn the mapping from the generated masks to real images [63]. Similarly, [39] is a nuclei segmentation method, where an initial segmentation model is used to generate weak labels for the test set. Then, an image-to-image translation model is trained to map the mask to the microscopy images. The synthetic masks are created by sampling nuclei instances from a mask database, then placing them at random positions on

the canvas. Generating synthetic masks with a GAN then using a conditional GAN to synthesize the corresponding microscopy image is also exploited [79]. Although this approach is similar to ours, there are major differences: (1): the method only generates binary masks with fragmented and clustered synthetic ground truth instances. In contrast, our mask representation generates real instances using GANs where the objects can share boundaries to challenge the downstream instance segmentation networks. (2): the method uses U-Net [88] as a downstream solver and uses pixel level metric, while our approach is designed for instance segmentation to match instances.Cross-modality image style transfer can also be used to enrich the underrepresented modalities in a multimodal dataset [56]. After clustering the images into 6 types using K-means based on color features extracted from the HSV space, a multimodal unpaired image-to-image translation task (DRIT [52]) is solved: the images are first encoded with the domain invariant content encoder network that captures the nuclei information from the images, while the attribute encoder captures the image style. Using the latent vectors coming from the content encoder and the styles extracted using the attribute encoder, synthetic images with different modalities can be generated with the same content (nuclei positions) to compensate for the modality imbalance in the dataset. On the synthetic images, the top submissions from the DSB 2018 competition are tested. The latter method is similar to ours as it also targets an instance segmentation problem but represents the content implicitly (as latent vectors), while our proposed method explicitly generates the synthetic masks and then uses (paired) image-to-image translation to generate the corresponding images. Table 8.1 summarizes the methods.

8.3 Datasets

We used two single-cell datasets for testing the proposed method (Figure 8.1). One is a salivary gland tumor dataset referred to as salivary gland (the nuclei and a tumor marker IHC stained) [74], while the other is a fluorescently stained fallopian tube tissue (membrane and nuclei staining) named fallopian [74]. In both of the datasets, the objects follow a specific layout that could be hard to explicitly generate using classic algorithms but using our approach, we can generate samples that are closer to the training dataset. (Figure 8.5) We also show qualitatively that using the StyleGAN2-ada generator, we are able to draw synthetic samples from this distribution, without any explicit parameterization or algorithm. (Figure 8.4)

Salivary gland tumor dataset (salivary gland)

The dataset consists of 10 annotated 3-channel images with resolution of 600x800 and contains a total of 1058 labeled cells. 5+5 folds are formed (5 for each exper-

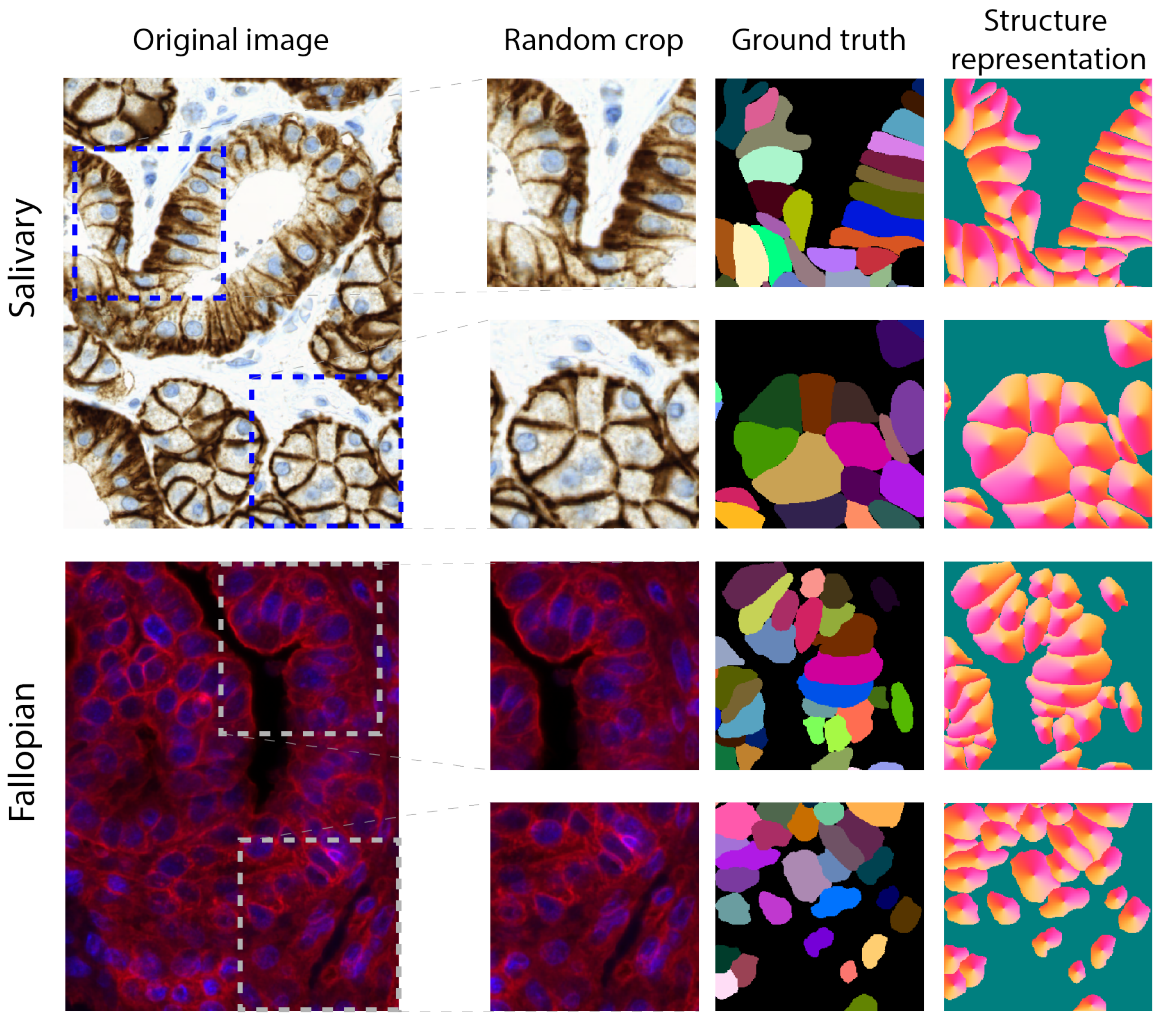


Figure 8.1: *Datasets: top: salivary gland, bottom: fallopian tube. The datasets are from the article [74]. The right side shows the input image crops, the corresponding labeled masks, and the structure representation of the masks first proposed in [99].*

Reference	Domain	Task	Unlabeled?	Metric
[10]	Medical images	Segmentation	No	DSC
[18, 90]	Medical images	Segmentation	Yes	DSC
[27, 28, 62, 100]	Medical images	Classification	No	ACC
[64]	Microscopy	Instance segmentation	Yes	Dice coeff.
[63]	Microscopy	Instance segmentation	No	AJI
[39]	Microscopy	Instance segmentation	Yes	DSB
[79]	Microscopy	Semantic segmentation	No	IoU
[56]	Microscopy	Instance segmentation	No	DSB
Proposed	**Microscopy**	**Instance segmentation**	**No**	**DSB**

Table 8.1: *A summary of the related works. The "Unlabeled?" value is "Yes" if the method uses the unlabeled samples (test set) as well. DSC means: **D**ice **S**imilarity **C**oefficient, ACC: classification accuracy, AJI: **A**verage **J**accard **I**ndex, DSB: Data Science Bowl 2018 standard metric [14] briefly introduced in subsection 8.5.3, IoU: **I**ntersection **o**ver **U**nion.*

iment), each fold contains 8 images for training and 1+1 images for validation and testing. (Figure 8.1, top)

Fallopian tube dataset (fallopian tube)

This dataset originally consisted of 8 images and 1818 annotated cells split into 30 parts with varying sizes. Only the parts reaching the resolution 256x256 pixels are kept, therefore we finally got 17 images. The number of cells are counted in each part and are distributed to 8 groups to each contain roughly equal numbers of annotated cells. From the 8 groups, 4+4 folds are formed, each fold contains 6 groups for training and 1+1 groups for validation and testing. (Figure 8.1, bottom)

8.4 Our method

Our method implements the GENERATE-SYNTHETIC-SAMPLES procedure. (Pseudocode 2) It operates on the training dataset $S = (I, M)$, $I = \{i_i\}_{i=1}^{N}$, $M = \{m_i\}_{i=1}^{N}$ of samples consisting of the input microscopy images and the corresponding segmentation masks. Each pixel in the segmentation image encodes the instance ID with the intensity, while the intensity 0 encodes the background.

ALGORITHM 2
The pseudocode of our method. It crops M overlapping patches from the input samples, then synthesizes and returns K samples that are assumed to be similar to the crops extracted from the N input images.

$S \leftarrow (\{i_i\}, \{m_i\})_{i=1}^{N}$
procedure GENERATE-SYNTHETIC-SAMPLES(S, K)
    $(\{i_j^c\}, \{m_j^c\})_{j=1}^{M} \leftarrow$ EXTRACT-CROPS(S))
    $m_{StyleGAN2-ada} \leftarrow$ STYLEGAN2-ADA-TRAIN(ENCODE($\{m_j^c\}$))
    $\{m_k^s\}_{k=1}^{K} \leftarrow$ DECODE(STYLEGAN2-ADA-GENERATE($m_{StyleGAN2-ada}, K$))
    $m_{pix2pix} \leftarrow$ PIX2PIX-TRAIN($\{m_j^c\}, \{i_j^c\}$)
    $\{i_k^s\} \leftarrow$ PIX2PIX-GENERATE($m_{pix2pix}, \{m_k^s\}$)
    return (i_k^s, m_k^s)
end procedure

8.4.1 Data preprocessing

We first extract overlapping crops of size 256x256 from the input images and masks using the EXTRACT-CROPS function. (Pseudocode 2.) A crop (with the corresponding image) is kept if it contains at least a few distinct objects, otherwise it is discarded. We also add the orthogonally rotated transformations of each crop to the dataset.

8.4.2 Synthetic mask generation

Generative Adversarial Networks (GANs). In traditional GANs the goal is to learn the mapping $G : z \rightarrow y$ where z is the element of the distribution of the training set. The learning employs two networks, the generator (G) and the discriminator (D), where both networks simultaneously improve to perform better in generating images (G) that can not be distinguished from the real images by the discriminator, and the discriminator trained to do that task more successfully. Thus, the value function is $V(G, D) = logD(y) + log(1 - D(G(z))$. V is minimized with respect to D and maximized with respect to G by doing one gradient update in each step. After the training has converged, G is used to generate samples from the distribution of the dataset and D is discarded. In the StyleGAN and its variants, the architecture of G is modified such that it progressively upsamples the image being generated while adds details to it. In each upsampling step, a style vector is used which is constructed by a multi-layered fully connected network from z.

Modeling the distribution of the instances. To generate the synthetic masks, we learn the distribution of the instances in the original dataset and then use the learned model to generate objects from the distribution Previous works generate the synthetic masks by sampling objects from a cell database and then placing them on an empty canvas after applying random transformations on them (rotation, resize, etc.)

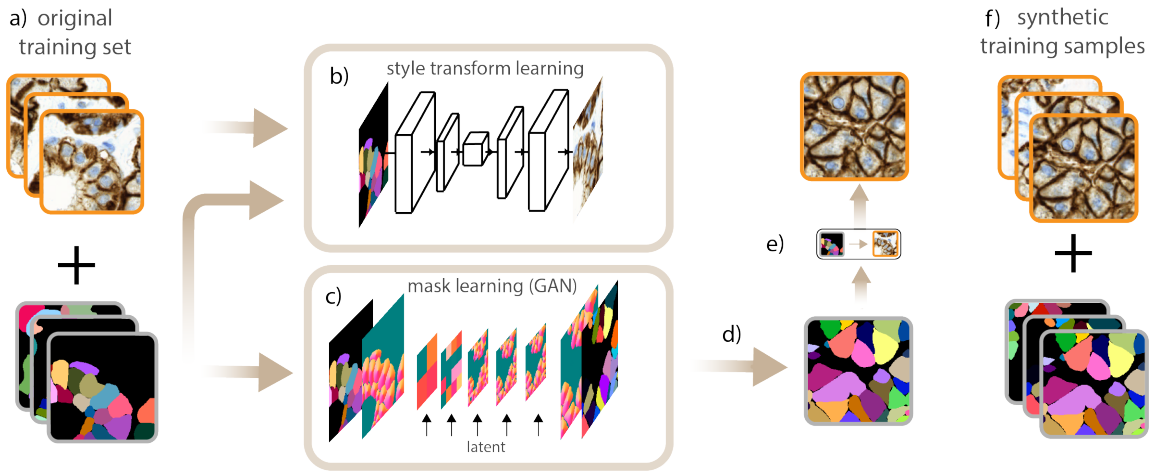


Figure 8.2: *The proposed model. a): the crops from the original dataset with the input microscopy images and their corresponding ground truth masks. b): learning an image-to-image translation model $m_{pix2pix}$ to translate the masks into the microscopy images in the training set. c): a StyleGAN2-ada model $m_{StyleGAN2-ada}$ is trained on the heat-flow representations of the masks in the training set. d): The learned GAN model is then used to generate heat-flows from the distribution learned from the training set which is then converted back into labeled masks. e): The learned style transform model is then applied on the synthetic masks. f): the synthetic dataset with the synthetic masks and the corresponding microscopy images (non-curated images). See pseudocode 2. for the details.*

[39, 63]. We argue that this way, we can not easily model complex distributions where the instances follow unique global structures as in our case (Figure 8.1). This is often the case in tissue samples. Therefore, we train a GAN on the masks to model not only the shape of the objects but their relative locations and orientations. As our method is intended to work on small annotated datasets, we have to train a GAN with a limited number of samples. Fortunately, recent GANs offer non-leaking data augmentation to learn from limited-size datasets [43]. In our proposed pipeline, we use the StyleGAN2-ada framework for all of the experiments: first a model $m_{StyleGAN2-ada}$ is learned using the STYLEGAN2-ADA-TRAIN function, then the synthetic samples are drawn using the learned model

Mask generation

A naive solution would be to feed the binarized version of the instance masks ($\{m_j^c\}$) directly into the GAN where all of the cells share the same label [79], while the background is marked with zeros (that is, the ENCODE function in pseudocode 1. is the threshold functional with threshold parameter 1). This strategy is suitable for tasks where only one or a few objects should be segmented, but in instance segmenta-

tion, especially in nuclei segmentation of tissues where objects densely located, this strategy has two main limitations: (1) instances that share boundaries with other instances are indistinguishable from each other, therefore touching cells can not be modeled. (2) we observed that if a model is trained on the binarized masks, the network generates lots of fragmented objects similarly to the presented graphical results where the authors also used binary mask representation [79].

We also experimented with the labeled masks where each object is encoded with a unique intensity value (the ENCODE function is identity). Although this way the generated cell instances are not fragmented, small intensity variance can be observed in almost each generated object that is nearly impossible to fix, and consequently, touching objects can not be modeled. This happens because the input has an inherently discrete property that is not respected by the StyleGAN2-ada as it learns the proper (continuous) distribution of pixels that results in perceptually appealing results, but no terms in the loss forces the dynamics of the learning to respect the discrete nature of the dataset. Results and common failure modes of these naive approaches are presented on Figure 8.3.

Structure representation

To overcome the issues above, we choose to encode our labeled masks into a dense and continuous structure representation to solve both problems. In theory, many representations could work (see the note below), but we found Cellpose's [99] heat flow simulation as the most robust for our task: first, the centroid of each cell is determined, and a constant heat is applied to that point in an iterative manner (Figure 8.1, right). The heat distribution is captured at the end of this iterative process, and the objects are represented using the gradients of the final heat distribution. Reconstruction is done by following the gradients in each pixel: if two different pixels converge to the same position, then they are representing the same object. We convert each mask into their vector-flow representation and feed them to the StyleGAN2-ada during training. The converted masks are encoded in three channels, where the first two channels represent the gradient of the flow (dx and dy) in each pixel, and the third channel encodes the object probability. The main advantage of this representation is that the instance masks can be represented as 3-channel images and there is no need to modify the StyleGAN2-ada to feed the masks in the vector-flow format. The vector-flow representation naturally solves the problem (1) since the pixels near the touching region converge to the reference points (centroids) of the objects they are part of. Based on our experiments, problem (2) is also solved, as we did not observe the fragmented objects in the generated vector-flows (Figure 8.3). After training the StyleGAN2-ada with the flows, we generate synthetic flows and decode them with the mentioned tracking algorithm.

Our method does not explicitly depend on the Cellpose representation. In theory,

any representation may work that can encode a labeled mask into a dense image. We also experimented with the gradient vector flow representation [107], but found that the Cellpose representation has higher tolerance on the inaccuracies generated by the GAN.

Generating the microscopy images from the masks

The pix2pix method solves the image-to-image translation task using modified conditional GANs. Compared to a regular conditional GAN, the pix2pix method adds the dependency on the condition not only to the generator but also the discriminator. Thus, the value function becomes $V(G, D) = logD(x, y) + log(1 - D(x, G(x, z))$, where the x is the condition (the source image), y is the target image and z is random latent vector. The loss function minimizes D and maximizes G by doing one gradient update for each input for both networks in each step. Once the training converges, the generator represents the mapping $(x, z) \rightarrow y$.

Parallel to training the StyleGAN2-ada model, we learn $m_{pix2pix}$ using image-to-image translation on the training set [41] that will be later used to transform the synthesized masks into their corresponding synthetic microscopy images. One can learn the mapping of the vector flow representation of the microscopy images directly (the first parameter of PIX2PIX-TRAIN is the raw output of STYLEGAN2-ADA-GENERATE) but we observed that learning the translation from the raw labeled masks into their corresponding microscopy images leads to better image quality on the datasets we are working on. After both the image-to-image translation task and the synthetic mask generation task is completed, the pix2pix model is used to translate the synthetic masks into the corresponding synthetic microscopy images, and both sets are returned. The whole pipeline is shown on Figure 8.2. and pseudocode 2.

8.4.3 Training the downstream task

The synthesized images and masks can be used to augment the training set used to generate the synthetic samples. We use the StarDist and Cellpose instance segmentation methods to demonstrate that the synthesized samples can improve the generalization capability of these networks. Although we tested with the networks above, our method does not depend on any particular instance segmentation method. In the following, we briefly introduce both methods.

StarDist instance segmentation

The masks in the training set are processed, and each object in the mask is converted into a star-convex polygon by first selecting the centroid of an object and then measuring the length of the rays connecting the contour of an object and its centroid. The

angle between adjacent rays is equal and fixed for the entire dataset. The network consists only of convolutional layers (“U-Net” and ”ResNet” are proposed), where the layer exactly before the top one branches to predict the probability map and the distance map. Both the probability and distance feature maps have spatial size proportional to the input size (or equal size if downsampling is not used). The probability map contains the object probability scores for each representative location, and the distance map at the same location represents the ray lengths encoding a candidate object (that is defined for the entire map). The training uses cross entropy loss to supervise the probability scores while uses mean squared error for the distances. During prediction, non maximum suppression is used to find the best candidates.

Cellpose

The method uses the heat-flow representation to encode each object in a labeled mask to a different image containing 3 channels: the probability map and a vector field encoding the flow. Since the heat-flow can be converted back to a labeled mask, the instance segmentation problem can be solved as a dense prediction task: the method uses a fully convolutional architecture (a “U-Net”-like network is used) to predict the flows that are converted back into labels.

8.5 Results

In this section, we first introduce how the mask generation fails when the intermediate representation is used (the GAN is trained on the binary or labeled masks directly). Then, we generate synthetic samples using the SIMCEP method and compare it to the proposed GAN based method. We quantitatively and qualitatively assess the synthetic image and mask quality and use the Fréchet Inception Distance (fID) metric in the former case. Lastly we trained a StarDist and a Cellpose model on each of the datasets to measure the improvement of the segmentation quality when the synthetic training samples are used.

8.5.1 Naive approaches for GAN training (ablation experiments)

We trained StyleGAN2-ada on binary and labeled masks (the ENCODE function is the thresholding or identity, respectively). The binary training is tested on the masks of DSB 2018 dataset, where only a few instances touch each other per image. We also tested the quality of the synthetic masks when the network trained on the raw labeled images of the salivary gland dataset. In the latter, the instances follow a denser layout (compared to the DSB 2018 containing mainly cell cultures), and most

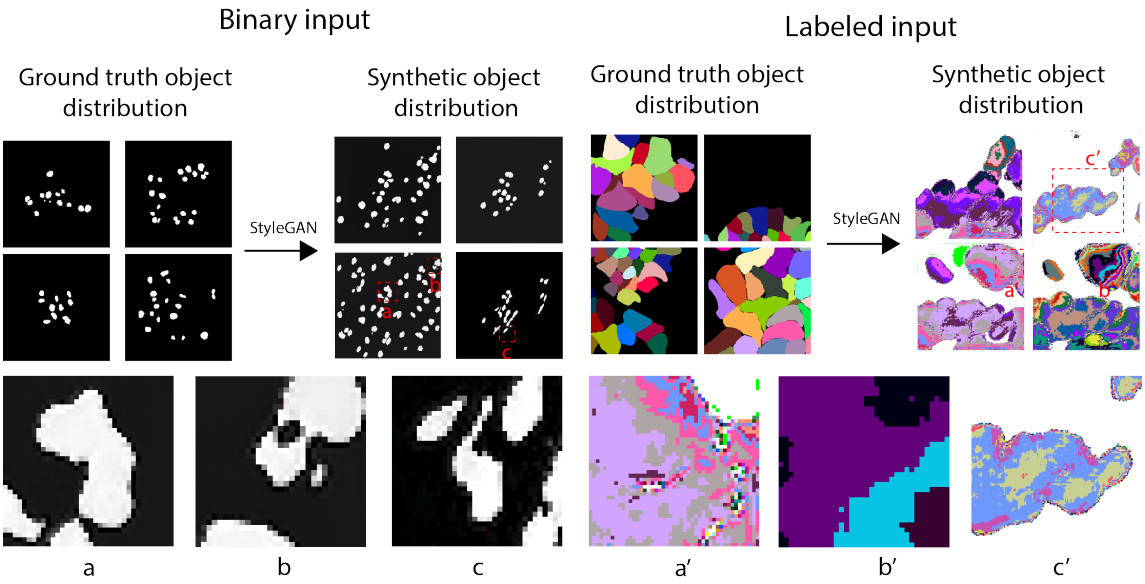


Figure 8.3: *Common errors when training StyleGAN2-ada directly on the masks. Top: the StyleGAN2-ada was trained on the binarized masks from the DSB2018 dataset. Bottom: trained on the labels of the salivary gland dataset directly. Common issues when training directly on the binarized masks: a: the objects can not be separated, the Stlyle-GAN generates blobs instead of instances. b: holes between the objects. c: fragmented objects with blurry boundaries. When training on the labels, the common issues are: a': holes in the objects, b': non-uniform intensities represent an object, therefore the reconstruction is nearly impossible. c': nuclei blobs, containing mixed intensities. (The input is grayscale in the labeled case, the colors are only added for better visualization).*

of the instances share boundaries with others. Figure 8.3 presents the most common failure modes when training the network with these strategies.

StyleGAN2-ada training

To train the StyleGAN2-ada model ($m_{StyleGAN2-ada}$), we used 256x256 pixel sized overlapping tiles extracted from each dataset. Only tiles containing at least 3 objects are kept.

The best model is selected based on **F**réchet **I**nception **D**istance (fID) computed based on the pretrained ImageNet weights [25, 36]. Although ImageNet is a natural image dataset, it is shown that the features extracted by the model on medical images are also meaningful to assess the generated image quality [43]. We observed that models with fID < 100 produce synthetic masks that are numerically correct (they can be reconstructed by following the gradients without any significant error in the reconstructed mask). As the training progresses, the fID score may decrease but the variability of the objects may become less diverse, therefore visual assessment may

also be needed.

In the case of the salivary gland dataset, the model used is trained from scratch, while the fallopian model is also tested by fine-tuning the salivary gland model for saving time and computational costs and also to demonstrate transfer capability of StyleGAN2-ada on the datasets.

Training from scratch. When training from scratch, the StyleGAN2-ada reaches the fID 54.33 at step 6500 on the salivary gland dataset, while the model converges on the fallopian tube after 12140 steps and reaches the lowest fID 56.45.

Transfer learning. When we used the checkpoint at step 5000 (5 million crops passed through the network) from the salivary gland model, we could fine-tune the network on the fallopian tube, and the network needs only 1800 steps (compared to the 12140 when trained from scratch) to reach the minimum score 46.61 that is also substantially better (17.4% lower distance from the ground truth compared to uninitialized training).

Fine-tuning with limited subsets. We used the salivary gland model at checkpoint 6400 (the best model on salivary gland) to fine tune on the crops extracted from a limited number of training images from the fallopian tube dataset. We observed that the model has a good enough score even when less than 50% of the training set is used. See the details in Supplementary material II: Table 2, 3.

8.5.2 Evaluation of the synthesized masks

Synthesized mask quality. We compare the masks generated using the SIMCEP method to our GAN based generation strategy by transforming each dataset (ground truth, generated by SIMCEP and ours) to the structure representation, and then compute the Fréchet Inception Distance (fID) of the generated datasets to the ground truth flows. We adjusted each possible hyperparameter of the SIMCEP to the parameters of the ground truth dataset (mean number of cells in each mask, min/max cell radius, estimated number of clusters) and generated a dataset of similar size to the ground truth. It is clearly demonstrated that the SIMCEP generator can achieve substantially better scores if the objects have simple structure but it fails to generate complex structures. This is obvious from the fID reached by the method and by visually inspecting the generated masks. When using the GAN based mask generation approach, the increase in the fID score is 3-fold on the salivary gland dataset and it is still 2x when comparing the fallopian masks to the ground truth. (Figure 8.4,8.5) This is not surprising as the salivary gland dataset has a richer cell structure layout. (Figure 8.1)

Synthesized microscopy image quality. We also compare the quality of the generated microscopy images when the learned style models are applied to the masks generated by SIMCEP and on the masks generated by our GAN based method. Again,

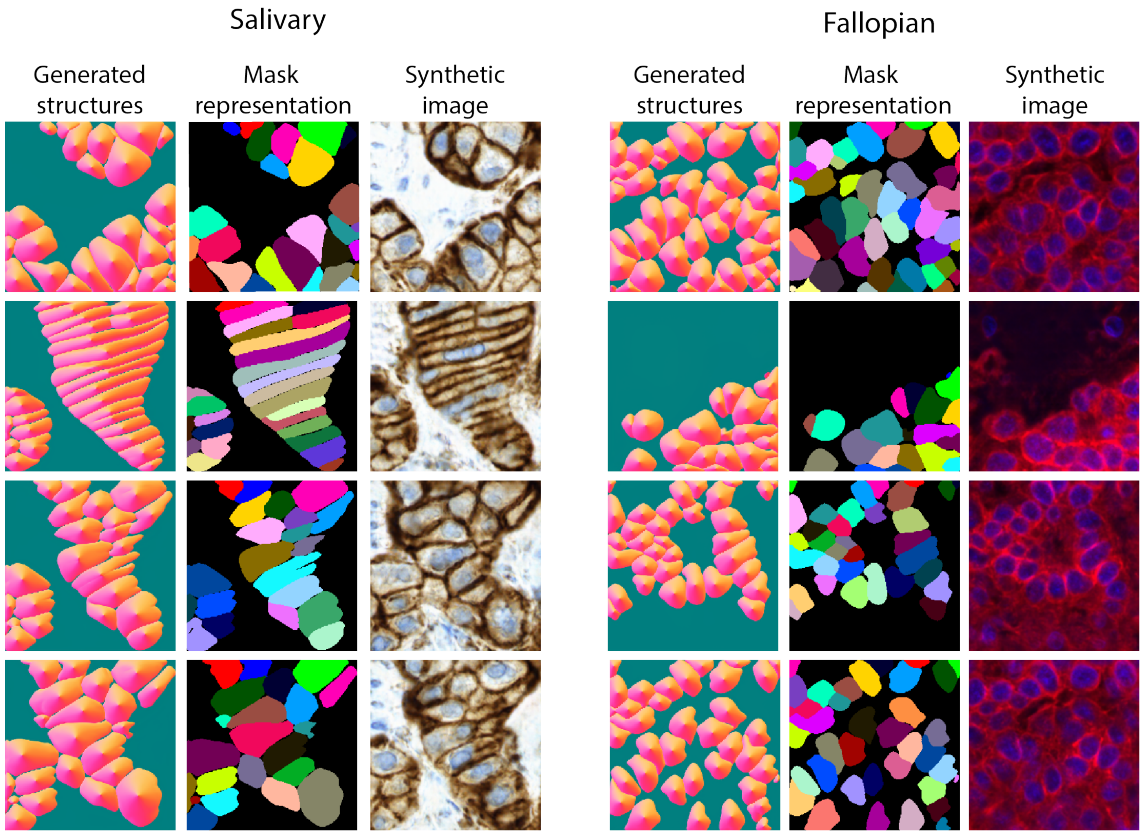


Figure 8.4: *Synthesized flows, the reconstructed masks and their corresponding microscopy images generated by our method.*

the fID score is substantially better on the fallopian dataset when using SIMCEP, as the cell structures found in this dataset are less complex. On the other hand, the fID scores are much worse when comparing the generated images from the salivary gland dataset to its ground truth. When using the masks generated by our method, a 2-fold increase can be observed in the fID scores for both datasets. (Figure 8.4,8.5)

Our quantitative and qualitative results (Figure 8.4,8.5) show that our approach is useful when the layout of the objects follows a complex distribution and it can not be easily approximated using the SIMCEP method. In contrast, the StyleGAN2-ada generator implicitly learns the distribution of the objects in the training set that can be used later to generate more realistic microscopy images compared to what can be achieved by utilizing the masks generated using the SIMCEP method.

To synthesize the microscopy images from the generated masks, we use the pix2pix [41] with default parameters found in the public repository of the method. The pix2pix models were trained for 600 and 300 epochs on the salivary gland and the fallopian, respectively. We used the result of the model of the last epoch to synthesize the microscopy images.

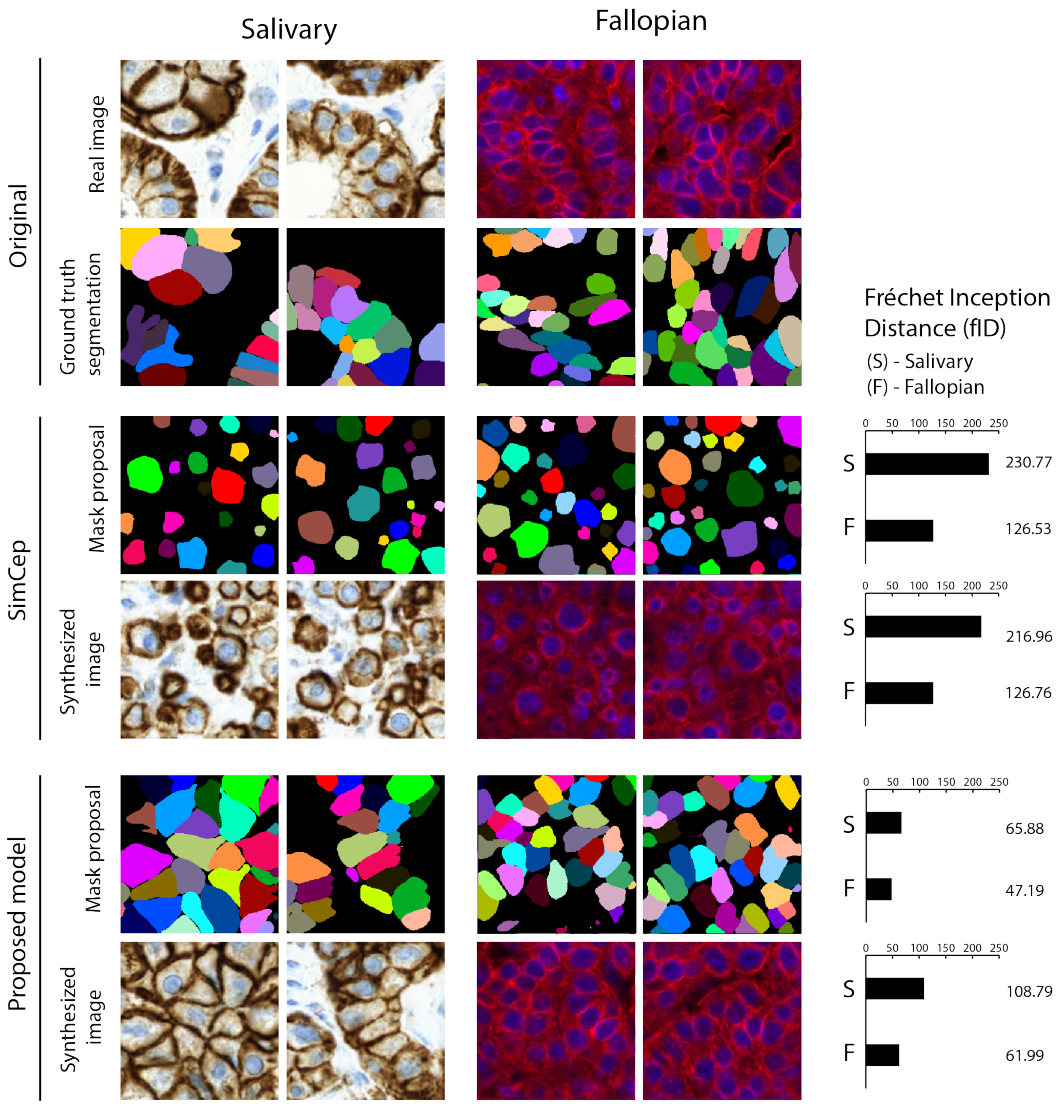


Figure 8.5: *Quantitative evaluation of the synthesized masks and their corresponding microscopy images generated by SIMCEP and the proposed model. The plot in the right column shows the Fréchet Inception Distance (fID) between the synthetic mask (microscopy image) and ground truth mask (microscopy image). From top to bottom: distance between ground truth mask and SIMCEP synthesized mask, distance between ground truth microscopy image and simulated microscopy image using pix2pix with SIMCEP synthesized mask input, distance between ground truth mask and StyleGAN2-ada synthesized masks, distance between ground truth microscopy image and simulated microscopy image using pix2pix with StyleGAN2-ada synthesized mask input.*

8.5.3 Instance segmentation results

We trained StarDist and Cellpose models on both datasets. Two experiments are performed. In the first, we pretrain the instance segmentation network on the synthesized images only, then the network is fine-tuned on the raw dataset. In the second experiment, we merged the raw dataset and the synthesized samples. We executed the pix2pix and instance segmentation network training with different training-set sizes to test the effectiveness of our method on an even more limited number of samples.

StarDist training: No-augmentation and pretrain models are trained through 100 epochs with batch size 4 and learning rate 0.003 and number of rays: 32. For the fine-tuning we limited the training for 20 epochs, as the models usually converged after only a few epochs. We used the last model from the pretraining as the initial weights in the transfer learning experiment.

Evaluation metrics. We used a standard nuclei segmentation metric to evaluate the performance of our model [14]. The metric matches the predicted and ground truth objects, and computes their intersection over union (IoU). Then the size of true positives, false positives and false negative sets are computed on each IoU threshold from 0.5 to 0.9 with step size 0.05. The metric then computes the mean accuracy (TP over TP + FP + FN) over the thresholds.

Transfer learning results

We trained StarDist and Cellpose models on both datasets. Tables 1 & 2 show the instance segmentation results using the DSB 2018 metric. Each number is the average performance of the folds trained independently 3 times. The augmentations are applied on the fly. We do not use the elastic deformations on the Cellpose masks as the flows for the deformated masks should have been computed before each step that is computationally too expensive. (Table 8.2.)

We measure the generalization capability of our approach by also comparing it to basic augmentation pipelines. We used augmentations affecting only the input image and geometric transformations that affect both the input images and the corresponding masks. We apply random joint intensity change with coefficient sampled from uniform distribution $U(0.6, 2)$ and added bias sampled from $U(-0.2, 2)$ and apply standard Gaussian noise with strength 0.02. We observed that using this augmentation protocol alone degrades the performance in most of the cases. The interested reader can check the numbers in the Supplementary material. We apply random rotations and flips in each dimension with probability 0.5 independently. We also use elastic deformations applied in the original U-Net paper [88, 104].

	Salivary gland		Fallopian tube	
	StarDist	Cellpose	StarDist	Cellpose
raw training set	0.3443	0.4867	0.2484	0.3822
augmentation	0.3854	-	0.3310	-
fine tune + augmentation	0.3893	0.4876	0.3567	0.3864

Table 8.2: *Segmentation results with StarDist and Cellpose.*

		Fallopian tube / subsets			
network	synthetic	3	6	9	13
CP	+	0.2754 (+0.0434)	0.2906 (+0.062)	0.3132 (+0.0247)	0.3122 (+0.093)
CP	-	0.2320	0.2844	0.2885	0.3029
SD	+	0.2837 (+0.0930)	0.3056 (+0.0415)	0.3154 (+0.0256)	32.30 (+0.0162)
SD	-	0.1906	0.2641	0.2898	0.3068

Table 8.3: *Fallopian tube subset experiment results: the numbers are the mean of the accuracies computed on each fold. (The baseline numbers for the Cellpose are significantly lower compared to Experiment 1, since we disabled the input uniformization for a more fair comparison with StarDist.)*

Subset experiments

We formed another 5 folds for both datasets and progressively eliminated images from the training set and synthesized the samples with the reduced dataset sizes. For the salivary gland dataset, we tested with 8 images (100% of the annotated images), 5 images (62,5%) and 3 images(37,5%). For the fallopian tube dataset, we considered 9 images as the 100% and used subsets of size 3 and 6. The segmentation task is then evaluated by merging the samples to the original dataset. We also observed, that the standard deviation of the accuracies of the repeated experiments on each fold is significantly lower, when the synthetic samples are used (see the supplementary for the actual numbers). (Tables 8.3 and 8.4.)

		Salivary gland / subsets		
network	synthetic	3	5	8
CP	+	0.3866 (+0.1261)	0.4215 (+0.1596)	0.4358 (+0.1062)
CP	-	0.2605	0.2619	0.3295
SD	+	0.2867 (+0.038)	0.3021 (+0.0378)	0.3079 (+0.0291)
SD	-	0.2487	0.2643	0.2788

Table 8.4: *Salivary gland subset experiment results: the numbers are the mean of the accuracies computed on each fold. (The baseline numbers for the Cellpose are significantly lower compared to Experiment 1, since we disabled the input uniformization for a more fair comparison with StarDist.)*

8.6 Summary

We introduced a synthetic sample generation strategy for instance segmentation which consists of the generation of the synthetic masks using a generative adversarial network (StyleGAN) and their corresponding synthetic microscopy images using image-to-image translation (pix2pix). Our methodology generates the labeled masks explicitly and they can also be used for other tasks as well. We proposed a general encoding technique and showed that the instances can be learned directly from the data using the StyleGAN2-ada. We showed that the distribution of the instances on the masks generated by the GAN are more similar to the ground truth distribution compared to the masks generated by classical methods like the SIMCEP, and the GAN generation is especially useful when the underlying structure of the objects are more complex than cell cultures like in our case. We qualitatively showed that the naive training of GANs on binary masks or the raw labels lead to suboptimal results as fragmented objects are often produced even in images from simple cell cultures and the reconstruction of the labels are almost impossible from the GAN output, but the mask encoding overcomes these issues. We also showed that the generated samples can be used to improve on the accuracy of the downstream instance segmentation task especially when only a very limited number of samples are available, compared to the case when only the raw dataset is used for training.

Chapter 9

Conclusion

In this thesis, we developed algorithms for microscopy image segmentation. The common in the developments is that we extract single cells, or their nucleus. Because we approached a problem with techniques from two subfields of computer science, we split the text into two parts. The first part focuses on active contour methods while the second part improves methods in deep learning.

In the active contour part (part I), we first concluded that implicit representation of the contour using the level set framework is preferred for implementation because of its multiple advantages. However, the level set framework has an important issue: durint the contour evolution, numerical errors are introduced and the level set deviates from the signed distance function.

In the first thesis point, we focused on solving this issue. We use the phase field functional that forms a smooth transition in the vicinity of the contour acting as a signed distance function locally. The phase field model, however, has a major issue: it contains a curvature dependent term, thus it significantly moves the contour after performing enough iterations. We experimentally showed that objects can disappear after applying phase field regularization many times even if the normal velocity is zero in the level set equation. We solved this issue by using an appropriate combination of the original gradient based term and a second order smoothness. The resulting differential equation can be implemented efficiently.

In the second thesis we develop the 3D extension of the selective active contour model. We show that we can retrieve objects selectively by using the volume prior and a shape prior that measures the complexity of the objects (sphericity). The model is combined with a region based data term that, for each surface point, measures the mean intensity differences in two cube shaped regions inside and outside of the surface aligned to the surface normal. The resulting 3D active contour model is embedded into the Medical Image Insight Toolkit (MITK) and can be used as an annotation tool. The user can manipulate the priors during the surface evolution to fit exactly to one object. Although the proposed software can decrease the annotation

time dramatically compared to fully manual 2D annotation, the software still needs lots of intervention from the user. The software did not exploit the combination of the geometrical models with data driven techniques. For example, the proposed data term considers intensity differences acting as a prior while an improved model might model the data term using a convolutional neural network. The idea of combining deep learning and geometrical models are not new. For example, [57, 81] model the contour as a Graph Convolutional Neural Network and in multiple forward step, the discretized contour approaches the target object. [40] models the velocity function to move the contour in the normal direction. Another method [55] moves the control points of a Catmull-Rom spline for interactive annotation. The user can move the control points of the spline and the new splines are predicted using the model. A recent preprint proposes a web based software to segment any object in an image by only a few clicks. [47] Such deep learning models are developed for mostly general image segmentation, however they do no incorporate prior information into the segmentation process that is one of the main strength of active contours derived from variational principles. A future work may successfully incorporates the advantages of both fields in a single model preferably in 3D.

In the deep learning part (part II), we introduced an automatic augmentation protocol for convolutional neural networks. Our first method, the *nucleAIzer* simulates instance masks containing nucleus objects using traditional cell population simulation methods (e.g. objects are drawn on an empty canvas to random locations drawn from a parametric distribution) and an image-to-image translation model is used to transform instance masks to microscopy images. The image-to-image translation model is trained on the test set where the instance masks are acquired using an instance segmentation model trained on external data. We showed that by using synthesized samples, the test accuracy is improved on the DSB 2018 [14] nuclei segmentation dataset compared to the cases when training on the initial dataset.

In a follow up research, we fixed the main issue of the previous model, and proposed a method to learn the instance masks from the data using a recent GAN model. The latter model is shown to be able to synthesize instance masks of microscopy images of complex cellular structures in tissue images (that would be hard to capture using parametric distributions). We showed on two tissue dataset that the synthesized samples are closer to the ground truth dataset by considering their Fréchet Inception Distances. We combined the initial dataset with the synthesized samples and showed with two instance segmentation networks that the augmentation protocol improves the test accuracy compared to the case when only the raw training dataset is used.

Although we and others observed increase in test accuracy by GAN based augmentation protocols, we also observed that the improvement is largely dependent on the synthesized microscopy image quality. Thus, the weakness of our results is that

we did not incorporate the test set in the loss function in the learning phase similarly to another method in the medical image analysis field. [18] The latter method tries to synthesize intensity difference maps and deformation fields using a GAN conditioned on the samples of the training set. The synthesized intensity difference map is added to the microscopy image, while the deformation field is applied on the mask as well. The GAN is trained to synthesize deformation fields and intensity maps that when applied on the input, the synthesized distribution is close to the test set. This result directly incorporates the test set into the learning process, while the nucleAIzer method exploits it through a form of semi supervised-learning. Considering a wider field, ideas from the neural architecture search may be borrowed. Recent method not only try to optimize the neural network architecture, but their hyperparameters and even the augmentation protocol. [85] Another method tries to find optimal augmentation policy using reinforcement learning [112] or by representation learning [108] although the improvement is not dramatic. Insights from these approaches might be a good future direction for searching augmentation policies automatically.

Appendix A

The Euler-Lagrange equation of the Euler elastica

Here we describe the derivation of the Euler-Lagrange equation associated with a more general problem: the arbitrary function of the sum curvature with Lagrangian $L = F(K)|\mathbf{N}|$, is provided. Basic knowledge of classical differential geometry is assumed.

Additional notations: The contravariant basis vectors are denoted by $\mathbf{S}^u$, $\mathbf{S}^v$ ($\mathbf{S}^i \cdot \mathbf{S}_k = \delta^i_k$, $\delta^i_k = 1$ if $i = k$, 0 otherwise). From here on $i, k, l \in \{u, v\}$. The direct (dyadic) product of two vectors $\mathbf{u}$, $\mathbf{v}$ is defined such that $(\mathbf{uv}) \cdot \mathbf{w} = \mathbf{u}(\mathbf{v} \cdot \mathbf{w})$. Metric and inverse metric components are denoted by $g_{ik} = \mathbf{S}_i \cdot \mathbf{S}_k$ and $g^{ik} = \mathbf{S}^i \cdot \mathbf{S}^k$ respectively. The Christoffel symbols for the embedded surfaces can be defined by $\Gamma^l_{ik} = \mathbf{S}^l \cdot \mathbf{S}_{ik}$, where $\mathbf{S}_{ik}$ being the second partial derivative of the position vector. The sum and the Gaussian curvatures as formulated by Gauss are

$$K = \frac{g_{vv}(\mathbf{S}_{uu} \cdot \mathbf{n}) - 2g_{uv}(\mathbf{S}_{uv} \cdot \mathbf{n}) + g_{vv}(\mathbf{S}_{vv} \cdot \mathbf{n})}{|\mathbf{N}|^2}, \tag{A.1}$$

$$K_G = \frac{(\mathbf{S}_{uu} \cdot \mathbf{n})(\mathbf{S}_{vv} \cdot \mathbf{n}) - (\mathbf{S}_{uv} \cdot \mathbf{n})^2}{|\mathbf{N}|^2}, \tag{A.2}$$

where the denominator is the square of the normal vector $\mathbf{N} = \mathbf{S}_u \times \mathbf{S}_u$ and $|\mathbf{N}|^2$ is the determinant of the metric $|\mathbf{N}|^2 = g_{uu}g_{vv} - g^2_{uv}$. The basic differential geometry formulae used for the derivation of the Euler Lagrange equation are collected below. An arbitrary vector $\mathbf{w} \in \mathbb{R}^3$ can be decomposed at the surface points using either the covariant or the contravariant basis as

$$\begin{aligned} \mathbf{w} &= w^u \mathbf{S}_u + w^v \mathbf{S}_v + w^\perp \mathbf{n} \\ \mathbf{w} &= w_u \mathbf{S}^u + w_v \mathbf{S}^v + w^\perp \mathbf{n}, \end{aligned} \tag{A.3}$$

where $w^i = \mathbf{S}^i \cdot \mathbf{w}$, $w_i = \mathbf{S}_i \cdot \mathbf{w}$ and $w^\perp = \mathbf{n} \cdot \mathbf{w}$. The relations between the contravariant

and covariant basis are

$$\begin{aligned}\mathbf{S}_i &= g_{iu}\mathbf{S}^u + g_{iv}\mathbf{S}^v \\ \mathbf{S}^i &= g^{iu}\mathbf{S}_u + g^{iv}\mathbf{S}_v.\end{aligned} \tag{A.4}$$

Note that the contravariant basis can also be expressed by the following cross products:

$$\mathbf{S}^u = \frac{1}{|\mathbf{N}|}\mathbf{S}_v \times \mathbf{n},\ \mathbf{S}^v = \frac{1}{|\mathbf{N}|}\mathbf{n} \times \mathbf{S}_v. \tag{A.5}$$

The second partial derivatives of the position can be decomposed as vector is

$$\mathbf{S}_{ik} = \Gamma^u_{ik}\mathbf{S}_u + \Gamma^v_{ik}\mathbf{S}_v + (\mathbf{S}_{ik}\cdot\mathbf{n})\,\mathbf{n}, \tag{A.6}$$

where $\mathbf{I} = \mathbf{S}^u\mathbf{S}_u + \mathbf{S}^v\mathbf{S}_v + \mathbf{nn}$ being the identity tensor ($\mathbf{I}\cdot\mathbf{w} = \mathbf{w}\cdot\mathbf{I} \equiv \mathbf{w}$, $\mathbf{w} \in \mathbb{R}^3$). From (A.6)

$$\mathbf{S}_{ik}\cdot\mathbf{S}_l = \Gamma^u_{ik}g_{ul} + \Gamma^v_{ik}g_{vl}. \tag{A.7}$$

The (right) gradient of any quantity $\mathbf{X}$ restricted to the surface is defined by $\mathbf{X}\nabla \doteq \frac{\partial\mathbf{X}}{\partial u}\mathbf{S}^u + \frac{\partial\mathbf{X}}{\partial v}\mathbf{S}^v$. The divergence of a vector field $\mathbf{Y}$ is defined by $\mathbf{Y}\cdot\nabla \doteq \frac{\partial\mathbf{Y}}{\partial u}\cdot\mathbf{S}^u + \frac{\partial\mathbf{Y}}{\partial v}\cdot\mathbf{S}^v$. Simple calculation shows that the sum curvature can be expressed as the negative of the divergence of the unit normal vector:

$$K = -\mathbf{n}_u\cdot\mathbf{S}^u - \mathbf{n}_v\cdot\mathbf{S}^v = \mathbf{n}\cdot\mathbf{S}^u_u + \mathbf{n}\cdot\mathbf{S}^v_v. \tag{A.8}$$

The following basic identities are directly follow from the definitions:

$$\frac{\partial g_{uk}}{\partial\mathbf{S}_u} = \mathbf{S}_k \tag{A.9}$$

$$\frac{\partial\,|\mathbf{N}|}{\partial\mathbf{S}_k} = |\mathbf{N}|\,\mathbf{S}^k \tag{A.10}$$

$$\frac{\partial\mathbf{S}_{ik}\cdot\mathbf{N}}{\partial\mathbf{S}_u} = |\mathbf{N}|\left[(\mathbf{S}_{ik}\cdot\mathbf{n})\,\mathbf{S}^u - \Gamma^u_{ik}\mathbf{n}\right], \tag{A.11}$$

e.g. for (A.10), formulae (A.5) are used. The partial derivatives of $\mathbf{n}$ and$|\mathbf{N}|$ (from (A.10)) are:

$$\mathbf{n}_k = -\left(\mathbf{n}\cdot\mathbf{S}_{uk}\right)\mathbf{S}^u - \left(\mathbf{n}\cdot\mathbf{S}_{vk}\right)\mathbf{S}^v \tag{A.12}$$

$$|\mathbf{N}|_k = |\mathbf{N}|\left(\Gamma^u_{uk} + \Gamma^v_{vk}\right). \tag{A.13}$$

In (A.12), decomposition (A.3) and $\mathbf{S}_i\cdot\mathbf{n} \equiv 0 \to \mathbf{n}_k\cdot\mathbf{S}_i = -\mathbf{n}\cdot\mathbf{S}_{ik}$, in (A.13) identities $|\mathbf{N}|_k = \frac{\partial|\mathbf{N}|}{\partial\mathbf{S}_u}\cdot\mathbf{S}_{uk} + \frac{\partial|\mathbf{N}|}{\partial\mathbf{S}_v}\cdot\mathbf{S}_{vk}$ are used. Starting with an equivalent expression to (A.1) $K = K\frac{|\mathbf{N}|}{|\mathbf{N}|}$ (to replace $\mathbf{S}_{ik}\cdot\mathbf{n}$ with $\mathbf{S}_{ik}\cdot\mathbf{N}$ in the numerator) and using formulae

(A.9,A.10):

$$\begin{aligned}\frac{\partial K}{\partial \mathbf{S}_u} =& 2\frac{(\mathbf{S}_{vv}\cdot\mathbf{n})\,\mathbf{S}_u-(\mathbf{S}_{uv}\cdot\mathbf{n})\,\mathbf{S}_v}{|\mathbf{N}|^2}\\ &-\frac{g_{vv}\Gamma^u_{uu}-2g_{uv}\Gamma^u_{uv}+g_{uu}\Gamma^u_{vv}}{|\mathbf{N}|^2}\mathbf{n}\\ &-2K\mathbf{S}^u.\end{aligned} \tag{A.14}$$

Note that applying the first line of (A.4), the (half of the) first term of (A.14) can be alternatively written as

$$\begin{aligned}&\frac{g_{uu}\,(\mathbf{S}_{vv}\cdot\mathbf{n})-g_{uv}\,(\mathbf{S}_{uv}\cdot\mathbf{n})}{|\mathbf{N}|^2}\mathbf{S}^u\\ &+\frac{g_{uv}\,(\mathbf{S}_{vv}\cdot\mathbf{n})-g_{vv}\,(\mathbf{S}_{uv}\cdot\mathbf{n})}{|\mathbf{N}|^2}\mathbf{S}^v.\end{aligned} \tag{A.15}$$

The Euler-Lagrange equation: the equation for the Lagrangian having second order derivatives can be arranged as:

$$\begin{aligned}&\frac{\partial}{\partial u}\left(-\frac{\partial L}{\partial \mathbf{S}_u}+\frac{\partial}{\partial u}\frac{\partial L}{\partial \mathbf{S}_{uu}}+\frac{1}{2}\frac{\partial}{\partial v}\frac{\partial L}{\partial \mathbf{S}_{uv}}\right)\\ &+\frac{\partial}{\partial v}\left(-\frac{\partial L}{\partial \mathbf{S}_v}+\frac{\partial}{\partial v}\frac{\partial L}{\partial \mathbf{S}_{vv}}+\frac{1}{2}\frac{\partial}{\partial u}\frac{\partial L}{\partial \mathbf{S}_{uv}}\right).\end{aligned} \tag{A.16}$$

The calculations for the first three terms are as follows: a) $\frac{\partial L}{\partial \mathbf{S}_u}=|\mathbf{N}|\frac{\partial F}{\partial K}\frac{\partial K}{\partial \mathbf{S}_u}$; expanding the right side

$$\begin{aligned}\frac{\partial L}{\partial \mathbf{S}_u} =& |\mathbf{N}|\left(F-2K\frac{dF}{dK}\right)\mathbf{S}^u\\ &+2\frac{dF}{dK}\frac{(\mathbf{S}_{vv}\cdot\mathbf{n})\,\mathbf{S}_u-(\mathbf{S}_{uv}\cdot\mathbf{n})\,\mathbf{S}_v}{|\mathbf{N}|}\\ &-\frac{dF}{dK}\frac{g_{vv}\Gamma^u_{uu}-2g_{uv}\Gamma^u_{uv}+g_{uu}\Gamma^u_{vv}}{|\mathbf{N}|}\mathbf{n},\end{aligned} \tag{A.17}$$

where (A.14) is used; b) $\frac{\partial}{\partial u}\frac{\partial L}{\partial \mathbf{S}_{uu}}+\frac{1}{2}\frac{\partial}{\partial u}\frac{\partial L}{\partial \mathbf{S}_{uu}}=\mathbf{S}_v\cdot\frac{\partial}{\partial u}\left(\frac{dF}{dK}\frac{\mathbf{S}_v}{|\mathbf{N}|}\mathbf{n}\right)-\mathbf{S}_u\cdot\frac{\partial}{\partial v}\left(\frac{dF}{dK}\frac{\mathbf{S}_v}{|\mathbf{N}|}\mathbf{n}\right)$,

using $\frac{\partial L}{\partial \mathbf{S}_{uu}} = \frac{dF}{dK}\frac{g_{vv}}{|\mathbf{N}|}\mathbf{n}$, $\frac{1}{2}\frac{\partial L}{\partial \mathbf{S}_{uv}} = -\frac{dF}{dK}\frac{g_{uv}}{|\mathbf{N}|}\mathbf{n}$, and expanding the right side

$$\begin{aligned}
&\frac{\partial}{\partial u}\frac{\partial L}{\partial \mathbf{S}_{uu}} + \frac{1}{2}\frac{\partial}{\partial u}\frac{\partial L}{\partial \mathbf{S}_{uu}} \\
&\quad = \frac{d^2F}{dK^2}\left(g_{vv}\frac{\partial K}{\partial u} - g_{uv}\frac{\partial K}{\partial v}\right)\frac{\mathbf{n}}{|\mathbf{N}|} \\
&\qquad + \frac{dF}{dK}\frac{2g_{uv}\Gamma^u_{uv} - g_{uu}\Gamma^u_{vv} - g_{vv}\Gamma^u_{uu}}{|\mathbf{N}|}\mathbf{n} \\
&\qquad + \frac{dF}{dK}\frac{-g_{vv}\left(\mathbf{S}_{uu}\cdot\mathbf{n}\right) + g_{uv}\left(\mathbf{S}_{uv}\cdot\mathbf{n}\right)}{|\mathbf{N}|}\mathbf{S}^u \\
&\qquad + \frac{dF}{dK}\frac{-g_{vv}\left(\mathbf{S}_{uv}\cdot\mathbf{n}\right) + g_{uv}\left(\mathbf{S}_{vv}\cdot\mathbf{n}\right)}{|\mathbf{N}|}\mathbf{S}^v,
\end{aligned} \tag{A.18}$$

where the second term is the sum given by the derivatives of the covariant basis vectors $\mathbf{S}_i$ (using (A.7)) and the denominator $|\mathbf{N}|$ (using (A.13)), whilst the third an fourth terms come from the derivatives of the unit normal vector $\mathbf{n}$ (using (A.12)). Adding (A.17,A.18) and using (A.15), the following terms remain:

$$\begin{aligned}
&\frac{\partial}{\partial u}\frac{\partial L}{\partial \mathbf{S}_{uu}} + \frac{1}{2}\frac{\partial}{\partial v}\frac{\partial L}{\partial \mathbf{S}_{uv}} - \frac{\partial L}{\partial \mathbf{S}_u} \\
&\quad = \frac{d^2F}{dK^2}\left(g_{vv}\frac{\partial K}{\partial u} - g_{uv}\frac{\partial K}{\partial v}\right)\frac{\mathbf{n}}{|\mathbf{N}|} \\
&\qquad + \frac{dF}{dK}\frac{\left(\mathbf{S}_{uv}\cdot\mathbf{n}\right)\mathbf{S}_v - \left(\mathbf{S}_{vv}\cdot\mathbf{n}\right)\mathbf{S}_u}{|\mathbf{N}|} \\
&\qquad + |\mathbf{N}|\left(K\frac{dF}{dK} - F\right)\mathbf{S}^u.
\end{aligned} \tag{A.19}$$

Applying same steps for the second three terms of (A.16) the result is identical to (A.19) with the indices u, v swapped. Summing up the terms in the normal direction, the Euler-Lagrange equation (A.16) takes the form

$$
\begin{aligned}
&|\mathbf{N}|\left(K\frac{dF}{dK}-F\right)\left(\mathbf{S}_u^u\cdot\mathbf{n}+\mathbf{S}_v^v\cdot\mathbf{n}\right)\\
&\quad+2\frac{dF}{dK}\frac{\left(\mathbf{S}_{uv}\cdot\mathbf{n}\right)^2-\left(\mathbf{S}_{uu}\cdot\mathbf{n}\right)\left(\mathbf{S}_{vv}\cdot\mathbf{n}\right)}{|\mathbf{N}|}\\
&\quad+\frac{d^3F}{dK^3}\left[\frac{g_{vv}}{|\mathbf{N}|}\left(\frac{\partial K}{\partial u}\right)^2-2\frac{g_{vv}}{|\mathbf{N}|}\frac{\partial K}{\partial u}\frac{\partial K}{\partial v}+\frac{g_{uu}}{|\mathbf{N}|}\left(\frac{\partial K}{\partial v}\right)^2\right]\\
&\quad+\frac{d^2F}{dK^2}\left[\frac{g_{vv}}{|\mathbf{N}|}\frac{\partial^2K}{\partial u^2}-2\frac{g_{vv}}{|\mathbf{N}|}\frac{\partial^2K}{\partial u\partial v}+\frac{g_{uu}}{|\mathbf{N}|}\frac{\partial^2K}{\partial v^2}\right]\\
&\quad+\frac{d^2F}{dK^2}\frac{\partial K}{\partial u}\left[-\frac{g_{vv}}{|\mathbf{N}|}\Gamma_{uu}^u+2\frac{g_{vv}}{|\mathbf{N}|}\Gamma_{uv}^u-\frac{g_{uu}}{|\mathbf{N}|}\Gamma_{vv}^u\right]\\
&\quad+\frac{d^2F}{dK^2}\frac{\partial K}{\partial v}\left[-\frac{g_{vv}}{|\mathbf{N}|}\Gamma_{uu}^v+2\frac{g_{vv}}{|\mathbf{N}|}\Gamma_{uv}^v-\frac{g_{uu}}{|\mathbf{N}|}\Gamma_{vv}^v\right].
\end{aligned}
\tag{A.20}
$$

Similar calculation shows that the components in the tangent plane are all zero. In (A.20) the first term includes the sum curvature (A.8), the second term the Gaussian curvature (A.2). Simple calculation shows that the sum of the last four lines is the ($|\mathbf{N}|$ times the) divergence of the gradient i.e. the Laplace-Beltrami of $\frac{dF}{dK}$: $\frac{\partial}{\partial u}\left[\left(\frac{\partial}{\partial u}\frac{dF}{dK}\right)\mathbf{S}^u+\left(\frac{\partial}{\partial v}\frac{dF}{dK}\right)\mathbf{S}^v\right]\cdot\mathbf{S}^u+\frac{\partial}{\partial v}\left[\left(\frac{\partial}{\partial u}\frac{dF}{dK}\right)\mathbf{S}^u+\left(\frac{\partial}{\partial v}\frac{dF}{dK}\right)\mathbf{S}^v\right]\cdot\mathbf{S}^v$ (the relation between the metric and inverse metric $\left[g^{ik}\right]=\left[g_{ik}\right]^{-1}$ is used). Finally the Euler-Lagrange equation associated with the Lagrangian $L=F(K)|\mathbf{N}|$ can be written as

$$
|\mathbf{N}|\left[\left(K\frac{dF}{dK}-F\right)K-2\frac{dF}{dK}K_G+\nabla\cdot\nabla\frac{dF}{dK}\right]\mathbf{n}=\mathbf{0},
\tag{A.21}
$$

where $\nabla\cdot\nabla$ is a usual notation for the Laplace-Beltrami operator. An alternative formula to (A.21) can be written as $|\mathbf{N}|\left[\left(K\frac{dF}{dK}-F\right)K-2\frac{dF}{dK}K_G+\left(\frac{dF}{dK}\nabla\nabla\right)\cdot\cdot\mathbf{I}^{-1}\right]\mathbf{n}=\mathbf{0}$, where $\mathbf{I}$ stands for the first fundamental form i.e. the metric tensor with components g_{ik} and "$\cdot\cdot$" is the double scalar product operator (the result is the sum of the products of the corresponding components of the tensors involved in the operation). Using similar steps, one can deduce Euler-Lagrange equation for the arbitrary function of the Gaussian curvature, that turns to be

$$
|\mathbf{N}|\left[2K\left(F-\frac{dF}{dK_G}\right)+K_G\left(\frac{dF}{dK_G}\nabla\nabla\right)\cdot\cdot\mathbf{II}^{-1}\right]\mathbf{n}=\mathbf{0},
\tag{A.22}
$$

where $\mathbf{II}$ stands for the "second fundamental form" with components $\mathbf{S}_{ik}\cdot\mathbf{n}$.

Bibliography

[1] 2018 Data Science Bowl. `https://www.kaggle.com/c/data-science-bowl-2018`. Accessed: 2018-06-21.

[2] Michael D Abràmoff, Paulo J Magalhães, and Sunanda J Ram. Image processing with imagej. *Biophotonics international*, 11(7):36–42, 2004.

[3] David Adalsteinsson and James A Sethian. A fast level set method for propagating interfaces. *Journal of computational physics*, 118(2):269–277, 1995.

[4] Vijay Badrinarayanan, Alex Kendall, and Roberto Cipolla. Segnet: A deep convolutional encoder-decoder architecture for image segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 39(12):2481–2495, 2017.

[5] Pascal Bamford and Brian Lovell. Unsupervised cell nucleus segmentation with active contours. *Signal processing*, 71(2):203–213, 1998.

[6] Andreas Bartschat, Eduard Hübner, Markus Reischl, Ralf Mikut, and Johannes Stegmaier. Xpiwit—an xml pipeline wrapper for the insight toolkit. *Bioinformatics*, 32(2):315–317, 2016.

[7] Stuart Berg, Dominik Kutra, Thorben Kroeger, Christoph N Straehle, Bernhard X Kausler, Carsten Haubold, Martin Schiegg, Janez Ales, Thorsten Beier, Markus Rudy, et al. Ilastik: interactive machine learning for (bio) image analysis. *Nature methods*, 16(12):1226–1232, 2019.

[8] Serge Beucher. Use of watersheds in contour detection. In *Proc. Int. Workshop on Image Processing, Sept. 1979*, pages 17–21, 1979.

[9] Serge Beucher. The watershed transformation applied to image segmentation. *Scanning Microscopy*, 1992(6):28, 1992.

[10] Christopher Bowles, Liang Chen, Ricardo Guerrero, Paul Bentley, Roger Gunn, Alexander Hammers, David Alexander Dickie, Maria Valdés Hernández, Joanna Wardlaw, and Daniel Rueckert. Gan augmentation: Augmenting training data using generative adversarial networks. *arXiv preprint arXiv:1810.10863*, 2018.

[11] Leo Breiman. Random forests. *Machine learning*, 45:5–32, 2001.

[12] Thomas Brox and Daniel Cremers. On local region models and a statistical interpretation of the piecewise smooth mumford-shah functional. *International journal of computer vision*, 84(2):184–193, 2009.

[13] Juan C Caicedo, Sam Cooper, Florian Heigwer, Scott Warchal, Peng Qiu, Csaba Molnar, Aliaksei S Vasilevich, Joseph D Barry, Harmanjit Singh Bansal, Oren Kraus, et al. Data-analysis strategies for image-based cell profiling. *Nature methods*, 14(9):849–863, 2017.

[14] Juan C Caicedo, Allen Goodman, Kyle W Karhohs, Beth A Cimini, Jeanelle Ackerman, Marzieh Haghighi, CherKeng Heng, Tim Becker, Minh Doan, Claire McQuin, et al. Nucleus segmentation across imaging experiments: the 2018 data science bowl. *Nature methods*, 16(12):1247–1253, 2019.

[15] Anne E Carpenter, Thouis R Jones, Michael R Lamprecht, Colin Clarke, In Han Kang, Ola Friman, David A Guertin, Joo Han Chang, Robert A Lindquist, Jason Moffat, et al. Cellprofiler: image analysis software for identifying and quantifying cell phenotypes. *Genome biology*, 7:1–11, 2006.

[16] Vicent Caselles, Francine Catté, Tomeu Coll, and Françoise Dibos. A geometric model for active contours in image processing. *Numerische Mathematik*, 66(1):1–31, 1993.

[17] Vicent Caselles, Ron Kimmel, and Guillermo Sapiro. Geodesic active contours. *IJCV*, 22(1):61–79, Feb 1997.

[18] Krishna Chaitanya, Neerav Karani, Christian F Baumgartner, Anton Becker, Olivio Donati, and Ender Konukoglu. Semi-supervised and task-driven data augmentation. In *Information Processing in Medical Imaging: 26th International Conference, IPMI 2019, Hong Kong, China, June 2–7, 2019, Proceedings 26*, pages 29–41. Springer, 2019.

[19] T. F. Chan, S. H. Kang, and J. Shen. Euler's Elastica and Curvature-Based Inpainting. *SIAM Journal on Applied Mathematics*, 63:564–592, 2002.

[20] Tony Chan and Luminita Vese. An active contour model without edges. In *Scale-Space Theories in Computer Vision: Second International Conference, Scale-Space'99 Corfu, Greece, September 26–27, 1999 Proceedings 2*, pages 141–151. Springer, 1999.

[21] Liang-Chieh Chen, George Papandreou, Iasonas Kokkinos, Kevin Murphy, and Alan L Yuille. Deeplab: Semantic image segmentation with deep convolutional

nets, atrous convolution, and fully connected crfs. *IEEE transactions on pattern analysis and machine intelligence*, 40(4):834–848, 2017.

[22] Dan Cireşan, Ueli Meier, Jonathan Masci, and Jürgen Schmidhuber. Multi-column deep neural network for traffic sign classification. *Neural networks*, 32:333–338, 2012.

[23] Ilaria De Santis, Ervin Tasnadi, Peter Horvath, Alessandro Bevilacqua, and Filippo Piccinini. Open-source tools for volume estimation of 3d multicellular aggregates. *Applied Sciences*, 9(8):1616, 2019.

[24] Natalie de Souza. Organoids. *Nature Methods*, 15(1):23–23, 2018.

[25] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*, pages 248–255. Ieee, 2009.

[26] Selim Esedog, Yen-Hsi Richard Tsai, et al. Threshold dynamics for the piecewise constant mumford–shah functional. *J. of Comp. Physics*, 211(1):367–384, 2006.

[27] Maayan Frid-Adar, Idit Diamant, Eyal Klang, Michal Amitai, Jacob Goldberger, and Hayit Greenspan. Gan-based synthetic medical image augmentation for increased cnn performance in liver lesion classification. *Neurocomputing*, 321:321–331, 2018.

[28] Maayan Frid-Adar, Eyal Klang, Michal Amitai, Jacob Goldberger, and Hayit Greenspan. Synthetic data augmentation using gan for improved liver lesion classification. In *2018 IEEE 15th international symposium on biomedical imaging (ISBI 2018)*, pages 289–293. IEEE, 2018.

[29] Kunihiko Fukushima. Neocognitron: A self-organizing neural network model for a mechanism of pattern recognition unaffected by shift in position. *Biological cybernetics*, 36(4):193–202, 1980.

[30] Pierre Geurts, Alexandre Irrthum, and Louis Wehenkel. Supervised learning with decision tree-based methods in computational and systems biology. *Molecular Biosystems*, 5(12):1593–1605, 2009.

[31] Ross Girshick, Jeff Donahue, Trevor Darrell, and Jitendra Malik. Rich feature hierarchies for accurate object detection and semantic segmentation. In *2014 IEEE Conference on Computer Vision and Pattern Recognition*, pages 580–587, 2014.

[32] Laurent Gole, Kok Haur Ong, Thomas Boudier, Weimiao Yu, and Sohail Ahmed. Opensegspim: a user-friendly segmentation tool for spim data. *Bioinformatics*, 32(13):2075–2077, 2016.

[33] Jaza Gul-Mohammed, Ignacio Arganda-Carreras, Philippe Andrey, Vincent Galy, and Thomas Boudier. A generic classification-based method for segmentation of nuclei in 3d images of early embryos. *BMC bioinformatics*, 15(1):1–12, 2014.

[34] Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross Girshick. Mask r-cnn. In *Proceedings of the IEEE international conference on computer vision*, pages 2961–2969, 2017.

[35] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016.

[36] Martin Heusel, Hubert Ramsauer, Thomas Unterthiner, Bernhard Nessler, and Sepp Hochreiter. Gans trained by a two time-scale update rule converge to a local nash equilibrium. *Advances in neural information processing systems*, 30, 2017.

[37] Tin Kam Ho. Random decision forests. In *Proceedings of 3rd international conference on document analysis and recognition*, volume 1, pages 278–282. IEEE, 1995.

[38] Reka Hollandi, Nikita Moshkov, Lassi Paavolainen, Ervin Tasnadi, Filippo Piccinini, and Peter Horvath. Nucleus segmentation: towards automated solutions. *Trends in Cell Biology*, 2022.

[39] Reka Hollandi, Abel Szkalisity, Timea Toth, Ervin Tasnadi, Csaba Molnar, Botond Mathe, Istvan Grexa, Jozsef Molnar, Arpad Balind, Mate Gorbe, et al. nucleaizer: a parameter-free deep learning framework for nucleus segmentation using image style transfer. *Cell Systems*, 10(5):453–458, 2020.

[40] Ping Hu, Bing Shuai, Jun Liu, and Gang Wang. Deep level sets for salient object detection. In *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 540–549, 2017.

[41] Phillip Isola, Jun-Yan Zhu, Tinghui Zhou, and Alexei A Efros. Image-to-image translation with conditional adversarial networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1125–1134, 2017.

[42] Molnar Jozsef, I. A. Szucs, Cs. Molnar, and P. Horvath. Active Contours for Selective Object Segmentation. In *WACV*, pages 1–9. IEEE Computer Society, 2016.

[43] Tero Karras, Miika Aittala, Janne Hellsten, Samuli Laine, Jaakko Lehtinen, and Timo Aila. Training generative adversarial networks with limited data. *Advances in neural information processing systems*, 33:12104–12114, 2020.

[44] Michael Kass, Andrew Witkin, and Demetri Terzopoulos. Snakes: Active contour models. *IJCV*, 1(4):321–331, 1988.

[45] R. Kimmel and A.M. Bruckstein. Regularized Laplacian Zero Crossings as Optimal Edge Integrators. *International Journal of Computer Vision*, 53(3):225–243, 2003.

[46] Ron Kimmel and Alfred M Bruckstein. Regularized laplacian zero crossings as optimal edge integrators. *IJCV*, 53(3):225–243, 2003.

[47] Alexander Kirillov, Eric Mintun, Nikhila Ravi, Hanzi Mao, Chloe Rolland, Laura Gustafson, Tete Xiao, Spencer Whitehead, Alexander C Berg, Wan-Yen Lo, et al. Segment anything. *arXiv preprint arXiv:2304.02643*, 2023.

[48] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. *Communications of the ACM*, 60(6):84–90, 2017.

[49] Harold W Kuhn. The hungarian method for the assignment problem. *Naval research logistics quarterly*, 2(1-2):83–97, 1955.

[50] Michael R Lamprecht, David M Sabatini, and Anne E Carpenter. Cellprofiler™: free, versatile software for automated biological image analysis. *Biotechniques*, 42(1):71–75, 2007.

[51] Yann LeCun, Bernhard Boser, John S Denker, Donnie Henderson, Richard E Howard, Wayne Hubbard, and Lawrence D Jackel. Backpropagation applied to handwritten zip code recognition. *Neural computation*, 1(4):541–551, 1989.

[52] Hsin-Ying Lee, Hung-Yu Tseng, Jia-Bin Huang, Maneesh Singh, and Ming-Hsuan Yang. Diverse image-to-image translation via disentangled representations. In *Proceedings of the European conference on computer vision (ECCV)*, pages 35–51, 2018.

[53] Antti Lehmussola, Pekka Ruusuvuori, Jyrki Selinummi, Heikki Huttunen, and Olli Yli-Harja. Computational framework for simulating fluorescence microscope images with cell populations. *IEEE transactions on medical imaging*, 26(7):1010–1016, 2007.

[54] Yupeng Li, Guo Cao, Tao Wang, Qiongjie Cui, and Bisheng Wang. A novel local region-based active contour model for image segmentation using bayes theorem. *Information Sciences*, 506:443–456, 2020.

[55] Huan Ling, Jun Gao, Amlan Kar, Wenzheng Chen, and Sanja Fidler. Fast interactive object annotation with curve-gcn. In *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 5252–5261, 2019.

[56] Ye Liu, Sophia J Wagner, and Tingying Peng. Multi-modality microscopy image style augmentation for nuclei segmentation. *Journal of Imaging*, 8(3):71, 2022.

[57] Zichen Liu, Jun Hao Liew, Xiangyu Chen, and Jiashi Feng. Dance : A deep attentive contour model for efficient instance segmentation. In *2021 IEEE Winter Conference on Applications of Computer Vision (WACV)*, pages 345–354, 2021.

[58] Jonathan Long, Evan Shelhamer, and Trevor Darrell. Fully convolutional networks for semantic segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3431–3440, 2015.

[59] William E Lorensen and Harvey E Cline. Marching cubes: A high resolution 3d surface construction algorithm. In *Seminal graphics: pioneering efforts that shaped the field*, pages 347–353. 1998.

[60] Xinghua Lou, Minjung Kang, Panagiotis Xenopoulos, Silvia Munoz-Descalzo, and Anna-Katerina Hadjantonakis. A rapid and efficient 2d/3d nuclear segmentation method for analysis of early mouse embryo and stem cell image data. *Stem cell reports*, 2(3):382–397, 2014.

[61] A. Witkin M. Kass and D. Terzopoulos. Snakes: Active Contour Models. *International Journal of Computer Vision*, 1(4):321–331, 1988.

[62] Ali Madani, Mehdi Moradi, Alexandros Karargyris, and Tanveer Syeda-Mahmood. Chest x-ray generation and data augmentation for cardiovascular abnormality classification. In *Medical imaging 2018: Image processing*, volume 10574, pages 415–420. SPIE, 2018.

[63] Faisal Mahmood, Daniel Borders, Richard J Chen, Gregory N McKay, Kevan J Salimian, Alexander Baras, and Nicholas J Durr. Deep adversarial training for multi-organ nuclei segmentation in histopathology images. *IEEE transactions on medical imaging*, 39(11):3257–3267, 2019.

[64] Michael Majurski, Petru Manescu, Sarala Padi, Nicholas Schaub, Nathan Hotaling, Carl Simon Jr, and Peter Bajcsy. Cell image segmentation using generative adversarial networks, transfer learning, and augmentations. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition workshops*, pages 0–0, 2019.

[65] Ravi Malladi, James A Sethian, and Baba C Vemuri. Shape modeling with front propagation: A level set approach. *IEEE TPAMI*, 17(2):158–175, 1995.

[66] Csaba Molnar, Ian H Jermyn, Zoltan Kato, Vesa Rahkama, Päivi Östling, Piia Mikkonen, Vilja Pietiäinen, and Peter Horvath. Accurate morphology preserving segmentation of overlapping cells based on active contours. *Scientific reports*, 6(1):1–10, 2016.

[67] Csaba Molnar, Zoltan Kato, and Ian Jermyn. A multi-layer phase field model for extracting multiple near-circular objects. In *Proceedings of the 21st International Conference on Pattern Recognition (ICPR2012)*, pages 1427–1430. IEEE, 2012.

[68] Csaba Molnar, Zoltan Kato, and Ian H Jermyn. A new model for the segmentation of multiple, overlapping, near-circular objects. In *2015 International Conference on Digital Image Computing: Techniques and Applications (DICTA)*, pages 1–5. IEEE, 2015.

[69] Jozsef Molnar, Csaba Molnar, and Peter Horvath. An object splitting model using higher-order active contours for single-cell segmentation. In *Advances in Visual Computing: 12th International Symposium, ISVC 2016, Las Vegas, NV, USA, December 12-14, 2016, Proceedings, Part I 12*, pages 24–34. Springer, 2016.

[70] Jozsef Molnar, Adam Istvan Szucs, Csaba Molnar, and Peter Horvath. Active contours for selective object segmentation. In *2016 IEEE Winter Conference on Applications of Computer Vision (WACV)*, pages 1–9. IEEE, 2016.

[71] Jozsef Molnar, Ervin Tasnadi, and Peter Horvath. A balanced phase field model for active contours. In Jan Lellmann, Martin Burger, and Jan Modersitzki, editors, *Scale Space and Variational Methods in Computer Vision*, pages 419–431, Cham, 2019. Springer International Publishing.

[72] Jozsef Molnar, Ervin Tasnadi, Balint Kintses, Zoltan Farkas, Csaba Pal, Peter Horvath, and Tivadar Danka. Active surfaces for selective object segmentation in 3d. In *2017 International Conference on Digital Image Computing: Techniques and Applications (DICTA)*, pages 1–7, 2017.

[73] D. Mumford. Elastica and Computer Vision. In Chandrajit L." Bajaj, editor, *Algebraic Geometry and its Applications: Collections of Papers from Shreeram S. Abhyankar's 60th Birthday Conference"*, pages 491–506, New York, NY, 1994. Springer New York.

[74] Andreas Mund, Fabian Coscia, András Kriston, Réka Hollandi, Ferenc Kovács, Andreas-David Brunner, Ede Migh, Lisa Schweizer, Alberto Santos, Michael Bzorek, et al. Deep visual proteomics defines single-cell identity and heterogeneity. *Nature Biotechnology*, 40(8):1231–1240, 2022.

[75] John Nickolls, Ian Buck, Michael Garland, and Kevin Skadron. Scalable parallel programming with cuda: Is cuda the parallel programming model that application developers have been waiting for? *Queue*, 6(2):40–53, 2008.

[76] Marco Nolden, Sascha Zelzer, Alexander Seitel, Diana Wald, Michael Müller, Alfred M Franz, Daniel Maleike, Markus Fangerau, Matthias Baumhauer, Lena Maier-Hein, et al. The medical imaging interaction toolkit: challenges and advances: 10 years of open-source development. *International journal of computer assisted radiology and surgery*, 8:607–620, 2013.

[77] Stanley Osher, Ronald Fedkiw, and K Piechor. Level set methods and dynamic implicit surfaces. *Appl. Mech. Rev.*, 57(3):B15–B15, 2004.

[78] Stanley Osher and James A Sethian. Fronts propagating with curvature-dependent speed: algorithms based on hamilton-jacobi formulations. *J. of Comp. Physics*, 79(1):12–49, 1988.

[79] Siddharth Pandey, Pranshu Ranjan Singh, and Jing Tian. An image augmentation approach using two-stage generative adversarial network for nuclei image segmentation. *Biomedical Signal Processing and Control*, 57:101782, 2020.

[80] Danping Peng, Barry Merriman, Stanley Osher, Hongkai Zhao, and Myungjoo Kang. A pde-based fast local level set method. *J. of Comp. Physics*, 155(2):410 – 438, 1999.

[81] Sida Peng, Wen Jiang, Huaijin Pi, Xiuli Li, Hujun Bao, and Xiaowei Zhou. Deep snake for real-time instance segmentation. In *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 8530–8539, 2020.

[82] Filippo Piccinini, Tamas Balassa, Antonella Carbonaro, Akos Diosdi, Timea Toth, Nikita Moshkov, Ervin A Tasnadi, and Peter Horvath. Software tools for 3d nuclei segmentation and quantitative analysis in multicellular aggregates. *Computational and structural biotechnology journal*, 18:1287–1300, 2020.

[83] Filippo Piccinini, Anna Tesei, Michele Zanoni, and Alessandro Bevilacqua. Revims: Software tool for estimating the volumes of 3-d multicellular spheroids imaged using a light sheet fluorescence microscope. *Biotechniques*, 63(5):227–229, 2017.

[84] Axel Poulet, Ignacio Arganda-Carreras, David Legland, Aline V Probst, Philippe Andrey, and Christophe Tatout. Nucleusj: an imagej plugin for quantifying 3d images of interphase nuclei. *Bioinformatics*, 31(7):1144–1146, 2015.

[85] Pengzhen Ren, Yun Xiao, Xiaojun Chang, Po-Yao Huang, Zhihui Li, Xiaojiang Chen, and Xin Wang. A comprehensive survey of neural architecture search: Challenges and solutions. *ACM Computing Surveys (CSUR)*, 54(4):1–34, 2021.

[86] Marie Rochery, Ian Jermyn, and Josiane Zerubia. Phase field models and higher-order active contours. In *Computer Vision, 2005. ICCV 2005. Tenth IEEE International Conference on*, volume 2, pages 970–976. IEEE, 2005.

[87] Remi Ronfard. Region-based strategies for active contour models. *International journal of computer vision*, 13(2):229–251, 1994.

[88] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. In *Medical Image Computing and Computer-Assisted Intervention–MICCAI 2015: 18th International Conference, Munich, Germany, October 5-9, 2015, Proceedings, Part III 18*, pages 234–241. Springer, 2015.

[89] Nestor Saiz, Minjung Kang, Nadine Schrode, Xinghua Lou, and Anna-Katerina Hadjantonakis. Quantitative analysis of protein expression to study lineage specification in mouse preimplantation embryos. *JoVE (Journal of Visualized Experiments)*, (108):e53654, 2016.

[90] Veit Sandfort, Ke Yan, Perry J Pickhardt, and Ronald M Summers. Data augmentation using generative adversarial networks (cyclegan) to improve generalizability in ct segmentation tasks. *Scientific reports*, 9(1):16884, 2019.

[91] Johannes Schindelin, Ignacio Arganda-Carreras, Erwin Frise, Verena Kaynig, Mark Longair, Tobias Pietzsch, Stephan Preibisch, Curtis Rueden, Stephan Saalfeld, Benjamin Schmid, et al. Fiji: an open-source platform for biological-image analysis. *Nature methods*, 9(7):676–682, 2012.

[92] Uwe Schmidt, Martin Weigert, Coleman Broaddus, and Gene Myers. Cell detection with star-convex polygons. In *Medical Image Computing and Computer Assisted Intervention–MICCAI 2018: 21st International Conference, Granada, Spain, September 16-20, 2018, Proceedings, Part II 11*, pages 265–273. Springer, 2018.

[93] Caroline A Schneider, Wayne S Rasband, and Kevin W Eliceiri. Nih image to imagej: 25 years of image analysis. *Nature methods*, 9(7):671–675, 2012.

[94] Robert F Sekerka. Morphology: from sharp interface to phase field models. *Journal of crystal growth*, 264(4):530–540, 2004.

[95] James A Sethian. A fast marching level set method for monotonically advancing fronts. *proceedings of the National Academy of Sciences*, 93(4):1591–1595, 1996.

[96] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014.

[97] Christoph Sommer, Christoph Straehle, Ullrich Koethe, and Fred A Hamprecht. Ilastik: Interactive learning and segmentation toolkit. In *2011 IEEE international symposium on biomedical imaging: From nano to macro*, pages 230–233. IEEE, 2011.

[98] Rupesh K Srivastava, Klaus Greff, and Jürgen Schmidhuber. Training very deep networks. *Advances in neural information processing systems*, 28, 2015.

[99] Carsen Stringer, Tim Wang, Michalis Michaelos, and Marius Pachitariu. Cellpose: a generalist algorithm for cellular segmentation. *Nature methods*, 18(1):100–106, 2021.

[100] Shobhita Sundaram and Neha Hulkund. Gan-based data augmentation for chest x-ray classification. *arXiv preprint arXiv:2107.02970*, 2021.

[101] David Svoboda, Michal Kozubek, and Stanislav Stejskal. Generation of digital phantoms of cell nuclei and simulation of image formation in 3d image cytometry. *Cytometry Part A: The Journal of the International Society for Advancement of Cytometry*, 75(6):494–509, 2009.

[102] Ervin A Tasnadi, Timea Toth, Maria Kovacs, Akos Diosdi, Francesco Pampaloni, Jozsef Molnar, Filippo Piccinini, and Peter Horvath. 3d-cell-annotator: an open-source active surface tool for single-cell segmentation in 3d microscopy images. *Bioinformatics*, 36(9):2948–2949, 2020.

[103] Grétar Tryggvason, Bernard Bunner, Asghar Esmaeeli, Damir Juric, N Al-Rawahi, W Tauber, J Han, S Nas, and Y-J Jan. A front-tracking method for the computations of multiphase flow. *Journal of computational physics*, 169(2):708–759, 2001.

[104] Gijs van Tulder. elasticdeform: Elastic deformations for N-dimensional images, March 2021.

[105] Xuchu Wang, Jinxiao Shan, Yanmin Niu, Liwen Tan, and Shao-Xiang Zhang. Enhanced distance regularization for re-initialization free level set evolution with application to image segmentation. *Neurocomputing*, 141:223–235, 2014.

[106] David Wiesner, David Svoboda, Martin Maška, and Michal Kozubek. Cytopacq: a web-interface for simulating multi-dimensional cell imaging. *Bioinformatics*, 35(21):4531–4533, 2019.

[107] Chenyang Xu and Jerry L Prince. Gradient vector flow: A new external force for snakes. In *Proceedings of IEEE computer society conference on computer vision and pattern recognition*, pages 66–71. IEEE, 1997.

[108] Kaiwen Yang, Yanchao Sun, Jiahao Su, Fengxiang He, Xinmei Tian, Furong Huang, Tianyi Zhou, and Dacheng Tao. Adversarial auto-augment with label preservation: A representation learning principle guided approach. In *Advances in Neural Information Processing Systems*.

[109] Paul A Yushkevich, Joseph Piven, Heather Cody Hazlett, Rachel Gimpel Smith, Sean Ho, James C Gee, and Guido Gerig. User-guided 3d active contour segmentation of anatomical structures: significantly improved efficiency and reliability. *Neuroimage*, 31(3):1116–1128, 2006.

[110] D. Zhang and G. Lu. Review of shape representation and description techniques. *Pattern Recognition*, 37(1):1 – 19, 2004.

[111] Kaihua Zhang, Lei Zhang, Huihui Song, and David Zhang. Reinitialization-free level set evolution via reaction diffusion. *IEEE TIP*, 22(1):258–271, 2013.

[112] Xinyu Zhang, Qiang Wang, Jian Zhang, and Zhaobai Zhong. Adversarial autoaugment. *ArXiv*, abs/1912.11188, 2019.

Summary

Systems biology is an emerging interdisciplinary field that analyzes biological processes in a holistic way. Recent years, developments in microscopy technology have led to the possibility to acquire thousands of images of parallel experiments in a single day. The manual examination of such amount of data by experts is no longer practical. Therefore, efficient algorithms are needed to do the tasks instead of humans. In this thesis, we focused on the analysis of such microscopy images at single cell level using active contour methods (part I) and deep learning (part II).

Active contour methods and level sets

In fluorescent microscopy, cell components are painted using fluorescent dies that are excited using a light source emitting a particular wavelength of light and the resulting signal is recorded by a digital camera. Furthermore, confocal microscopes can image the biological sample at multiple depth levels with high axial resolution. This means, that one can acquire 2D images of a single nuclei instance at several e.g. 5-10 depth levels and construct a 3D volume of it. The motivation behind the 3D analysis of single cells is that it is believed that growing cell cultures in the 3D space (spheroids, organoids)[24] might be a much better model of living tissues that can be exploited in e.g. drug discovery to reduce the amount of experiments to be executed in animal models.

In chapter 4, we developed an active contour based method [72] to retrieve nucleus instances in 3D. We extended a 2D selective active contour model [70] that uses shape priors during the contour evolution. Such priors are the surface area, object volume, or the sphericity that is derived from the former two. We showed on synthetic and real microscopy experiments that using the proposed priors, we can selectively retrieve objects with different shapes.

For the numerical implementation, we represent the segmentation surface implicitly using the level set method [78]. The main advantage of the implicit representation is that the topology changes during the active contour evolution is handled automatically. One drawback, however, is that one has to periodically reinitialize the level set to signed distance function because of the numerical errors. In many cases,

approximate level set maintenance is sufficient, however, in our 3D selective active contour model, accurate reinitialization is needed to implement the shape prior. Accurate reinitialization needs the computation of exact Eucledian distances on the whole grid that makes the method inefficient in practice. To solve the problem, we proposed a local level set method [71] based on the phase field theory in chapter 3, that provides accurate and fast reinitialization. Our method is shown tho have minimal effect on the active contour model opposed to competing reinitialization methods.

In chapter 5, we combined the results of chapter 4 and chapter 3, combined a region based data term in 3D with the selective active contour model, and provided an efficient implementation of a 3D segmentation software called *3D-Cell-Annotator* [102]. The resulting tool allows the annotation of 3D images containing multiple single cells, even in cases, where multiple objects are touching each other. Although competing tools provide results almost instantly, even if their hyperparameters are optimized by us, their segmentation accuracy is significantly lower compared to what can be achieved by fully manual segmentation, or by the proposed annotation software. Compared to the fully manual annotation, our measurements show that the time needed for the annotation of a full image can be halved when it is done by the proposed tool and the segmentation accuracy remains similar.

Deep learning

In part II we propose automatic augmentation protocols for the nuclei segmentation of 2D microscopy images with convolutional neural networks.

Chapter 7 proposes a method called *nucleAIzer* [39] to enrich the microscopy images of rare modalities in the DSB 2018 nuclei segmentation dataset [14]. Our proposed method uses traditional parametric cell population simulation tools to simulate nuclei masks of microscopy images. Then, by using an instance segmentation model trained an external dataset, weak labels are generated to the test set, and an image-to-image translation model is trained to transform the weak nuclei masks into microscopy images. The synthesized nuclei masks are then transformed into microscopy images with the learned image-to-image translation model. We show that training on the combined set of the initial dataset and the synthesized samples, the instance segmentation accuracy on the test set of DSB 2018 increases.

In chapter 8 we present a follow up research for previous automatic augmentation algorithm. The main weakness of the nucleAIzer method is that the instance masks are synthesized using a traditional parametric cell population simulation method [53]. This is not a problem until we plan to segment nucleus instances of cell cultures as they can be approximated with ellipsoidal shapes and their location can be modelled with parametric distributions. We develop a method to simulate instance masks of

datasets where complex cell structures are observed. Our method is shown to be able to synthesize discrete instance masks of rich and complex cellular structures using a state of the art GAN. We train an image-to-image translation model on the training set to synthesize the corresponding microscopy images to the instance masks generated by the GAN. We show that the combination of the synthesized samples and the initial dataset leads to increased segmentation accuracy on the test set.

Author contributions

- Author contributions for thesis point I
  - I combined the proposed balanced phase field [BPF] model with different active contour models (geometric active contours, selective active contours in 2D and 3D), development of the numerical solution and their implementation (on GPU architecture & MATLAB).
  - Implementation of competing methods:
    - an improved accurate PDE based reinitialization method [RM] (Peng et al., 1999)
    - the Reaction-Diffusion reinitialization method [RD] (Zhang et al., 2013).
  - Design and execution of the empirical validation process:
  - Microscopy experiments: implemented the geodesic active contour & plugged in the balanced phase field model. Selected single cells from the DSB 2018 nuclei segmentation dataset, compared the BPF vs RD vs RM, computed peak segmentation accuracy and mean accuracy.
  - Synthetic experiments: generated synthetic binary masks, simulated the comparing methods (BPF vs RD vs RM) without any active contour model, measured the peak and mean IoU between the initial segmentation.
  - Re-executed the “dumbbell” experiment in the RD paper and tested with competing methods, evaluated and compared the movement of the (0-iso)contour.
- Author contributions for thesis point II
  - Adopted the theory of selective active contours from 2D (Molnar et al., 2016) to 3D. Developed the numerical solution for the selective active contour model in 3D on GPU architecture (except for the initial implementation that is: marching cubes surface extraction, working volume prior functional, Euler elastica and a simple edge based data term with OpenGL visualization)

- Working on the more problematic and complex shape prior, investigation of the convergence issues (that turned out it can be solved with accurate level set reinitialization), showed that it can efficiently solved by periodically minimizing the phase field functional (but causing side effects on the surface that is significantly reduced in thesis point I).
- Implementation of several level set reinitialization methods to be used when computing the shape prior:
  * Eikonal equation,
  * Chamfer-distance,
  * Eucledian (exact) reinitialization.
- Testing the numerical stability of the implemented level set reinitialization methods when the shape prior is used in the active contour model.
- Worked on a fast and efficient GPU connected components algorithm for object single object extraction.
- Virtual level set management (per object speed function) for selective object retrieval.
- Optimized local level set method and experimenting with narrow band techniques for efficient implementation on GPU.
- Synthetic experiment: testing the selectivity using the volume prior.
- Testing the selectivity on a real 3D microscopy image. Advising the acquisition of 3D images with a confocal microscope, selecting candidate regions that contained real yeast cells in different forms. Increased z-resolution by adding synthetic slices using optical flow. Prepared the figures for visualizing the surface evolution and volume rendering the microscopy image.
- Conceptualized & designed a 3D annotation process with the volume prior and the shape prior.
- Combined a 3D local region data term with the 3D selective active contour model, computing the fundamental quantities, implemented required matrix ops on GPU.
- Created a GPU dynamic library with a C++ interface.
- Created an MITK plugin that communicates with the GPGPU (general purpose GPU) library, surface rendering in MITK, surface visualization on the projection planes, multilabel segmentation support, segmentation import/export, live parameter updates on a GUI.
- Design and implementation of a contour-adaptive level set management procedure that minimizes the memory and computation needs for the an-

notation process. It can be greatly exploited when the objects to be segmented are small, but the image is significantly larger, that is the case in the segmentation of multicellular 3D cell cultures.

- Author contributions for thesis point III
  - I was responsible for the application of image-to-image translation (pix2pix) when developing our method (called nucleAIzer) for the Data Science Bowl competition.
  - Generated weak segmentations for the test set of the DSB 2018 dataset using a previously trained Mask R-CNN model.
  - Trained pix2pix models on the identified clusters.
  - Synthesized microscopy images for the given masks in each cluster using the learned pix2pix model.
  - Developed a Python package that measures the prediction accuracy.
  - Developed a single script that executes the elements of the pipeline.
  - Took part in the development of the online interface `https://nucleaizer.org`.
  - I developed an open source plugin for the Napari image analysis software `https://www.napari-hub.org/plugins/napari-nucleaizer`.
  - Identified the main weakness of the nucleAIzer method and conceptualized a new method that learns the discrete masks directly from the data.
  - Experimented with DCGAN to simulate binary masks.
  - Experimented with DCGAN and StarDist mask encoding.
  - Developed a GVF (Gradient Vector Flow) based encoding and decoding algorithm. Experimented with StyleGAN2-ada and the GVF-based encoding (results are not presented as it turned out that the Cellpose representation is more stable when decoding the generated images by the GAN).
  - Experimented with StyleGAN2-ada and Cellpose encoding that turned out to be the proper solution to synthesize discrete masks.
  - Identified two in-house dataset (tissues having complex structures) that can show the advantage of the GAN based instance mask generation namely salivary gland tumor and fallopian tube.
  - Synthesized instance masks with a classical parametric cell population simulation tool SIMCEP for comparison.
  - Modified the fID computation code in StyleGAN2-ada to compare two existing image datasets and measured the fID between the SIMCEP masks

	Thesis point		
	I	II	III
[1]			•
[2]			•
[3]		•	
[4]			•
[5]		•	
[6]		•	
[7]		•	
[8]	•		

Table A.1: *Connection between the author's publications and the thesis points. The author's publications are listed in a separate bibliography at the end of the thesis in the chapter "The author's relevant publications".*

& ground truth masks vs masks generated by our method and the ground truth masks.

– Experimented with the classical approach (learning the joint distribution of the masks and the microscopy images).

– Applied the pix2pix model that is trained on the initial dataset to synthesize microscopy images for the masks on both datasets.

– Synthesized microscopy images using the SIMCEP masks, showed that the quality is inferior to our method.

– Created cross-validation folds, executed the method on each fold 3 times with StarDist and Cellpose instance segmentation method to test if the method improves on the segmentation accuracy (when fine tuning the instance segmentation networks with the initial dataset trained on the synthesized samples only).

– Performed subset experiments: trained 10 independent models on each fold on multiple subsets of both datasets with both StarDist and Cellpose (subset experiments).

– Showed that the StyleGAN2-ada can be fine-tuned effectively from (converges after only feeding 1000kimg to the network) the salivary gland tumor to the fallopian tube dataset (that needs only a few hours compared to 1-2 days when trained from scratch).

– Empirically showed that using fine-tuning with limited subsets, the model can achieve relatively good fID.

Magyar nyelvű összefoglaló

A rendszerbiológia a közelmúltban rohamosan fejlődő interdiszciplináris terület az élettudományokban, mely a biológiai, élettani folyamatokat, rendszereket egészében vizsgálja. Az alkalmazott társtudományok a matematika, és részterületei, informatika, fizika, és mérnöki tudományok. A mikroszkópok rohamos fejlődésével lehetővé vált egyszerre több száz kísérlet párhuzamos lefolytatása, és azokról napi több ezer automatizált, nagy felbontású kép készítése, melyek lehetővé teszik a biológiai folyamatok egysejt-szintű vizsgálatát. A nagy mennyiségű képi adat elemzése azonban nem fizibilis megfelelő automatizáció nélkül. Jelen munkában olyan automatizált informatikai módszereket dolgoztunk ki, amelyekkel ezen felvételeket egysejt szinten tudjuk vizsgálni. A disszertáció logikailag két részre van bontva. Az I. részben a differenciálgeometrián alapuló aktív kontúr módszereket dolgotzunk ki, míg a II. részben a statisztikai alapú (deep learning) módszerekkel foglalkozunk.

Aktív kontúr modellek és a level set módszer

A fluoreszcens mikroszkópiában a sejtek egyes komponensei (például sejtmembrán, sejtmag) fluoreszcens festékekkel vannak megjelölve, amelyek bizonyos hullámhosszú fénnyel gerjesztve láthatóvá válnak a mikroszkópba épített digitális kamera számára. A konfokális mikroszkóptechnológiában történt fejlesztések ezen kívül lehetővé teszik azt, hogy a biológiai mintákról több mélységi szinten készítsünk felvételeket. A mikroszkópok ezen felbontása hasonló nagyságrendű lehet, mint a síkbeli felbontás. A különböző szinteken készített képeket sorrendben összefűzve ezek után 3-dimenziós képet kapunk, amelyen egy sejt akár 5-10 szeleten is látható. A motáviáció amögött, hogy 3 dimenzióban vizsgálunk képeket egysejt szinten az, hogy az utóbbi időkben kezdtek el elterjedni a 3 dimenziós sejtkultúrák. [24] Ezekben a biológiai modellekben a területen dolgozó szakértők azt feltételezik, hogy ha a 2 dimenziós sejtkultúrák helyett 3 dimenziós technikákat alkalmazunk, akkor azok jobban szimulálják az élelőlények szöveteiben végbemenő folyamatokat, így alkalmasak lehetnek egyes állatkísérletek kiváltására, így csökkentve például a gyógyszerkutatásokhoz szükséges kísérletek számát. Az I. részben az ilyen típusú felvételek példány alapú szegmentálásához fejlesztünk ki aktív kontúr alapú módszereket.

A 4. fejezetben egy 3D szelektív aktív kontúr alapú módszert [72] fejlesztettünk, arra a célra, hogy a 3 dimenziós képeken sejtmagok körvonalait meghatározzuk. A módszer a már létező 2-dimenziós szelektív aktív kontúr modell [70] 3D adaptációja: a szelektív modell lényege, hogy térfogat, felület, és alakleíró priorok (az előző kettő függvénye. gömbszerűség) alapján határozzuk meg azt, hogy milyen tulajdonságú objektumokat szeretnénk a képből kinyerni. Demonstráltuk szintetikusan generált képeken és egy valós mikroszkópos felvételen, hogy a priorok hogyan működnek.

A numerikus implementációhoz a level set módszert [78] használtuk, amelynek a legnagyobb előnye az, hogy az aktív kontúr evolúció közben a topológiát automatikusan kezeli szemben az energia minimalizálás alapú eredeti aktív kontúr modellel. A level set módszer egyik lényeges limitációja, hogy a kontúr indirekt reprezentálásához felhasznált level set függvény a differenciálegyenlet megoldása során fellépő numerikus hibák miatt egyre jobban elveszíti az előjeles távolságfüggvény (signed distance function) tulajdonságát. Mivel a kontúr sebességét meghatározó mennyiségeket (normálvektor, gradiens, görbület) a level set függvényből számítjuk ki, így ezen numerikus hibák hatással vannak magára a kontúrra. A megoldás a level set függvény rendszeres reinicializációja előjeles távolságfüggvénnyé. Ennek azonban az a hátránya, hogy a pontos távolságfüggvény kiszámítása időigényes feladat, a közelítő megoldások viszont nem minden esetben kielégítőek. A 3. fejezetben ezért egy phase field alapú megoldást dolgozunk ki [71], amely a kontúr környezetében természetes módon előállítja az előjeles távolságfüggvényt. A módszer egyetlen hátránya, hogy a phase field funkcionál minimalizálása közben a kontúr elmozdul, azaz nem neutrális. Ezt az elmozdulást a phase field Euler-Lagrange egyenletében lévő Laplace tag okozza. A módszerünk lényegében egy új másodrendű tag, és az eredeti gradiens-függő tag megfelelő kombinációjával kompenzálja ezt az elmozdulást.

Az 5. fejezetben a 4. fejezetben kifejlesztett aktív kontúr modellt kombináljuk a 3. fejezetben javasolt phase field reinicializációs módszerrel és egy félig automatizált szegmentáló módszert javaslunk 3 dimenziós felvételek annotálására. A módszer a legjobban az orvosi felvételek szegmentálására használatos Slicer aktív kontúr pluginjével állítható párhuzamba. A Slicer aktív kontúr pluginja azonban csak egy geodéziai aktív kontúr modellt tartalmaz, míg a mi módszerünk a szelektív aktív kontúr modellt javasolja a feladatra. Az utóbbinak az előnye, hogy a felhasználó az aktív kontúr evolúciót menet közben tudja befolyásolni a priorok és az egyenletben szereplő tagok súlyainak változtatásával. A szelektív modellt az MITK nevű orvosi képfeldolgozó szoftverbe ágyaztuk be, és nyílt forráskódú szoftverkéne publikáltunk *3D-Cell-Annotator* [102] néven.

Deep learning alapú módszerek

A dissszertáció II. része 2 dimenziós mikroszkópfelvételek szegmentálásával folgalkozik konvolúciós neurális hálózatokkal: olyan augmentációs eljárásokat mutatunk be, amelyekkel sikeresen növeltük a szegmentáció pontosságát a teszt halmazon. A két bemutatott eljárásban az a közös, hogy mindkét esetben az image-to-image translation módszerrel hozunk létre szintetikus mikroszkópos képeket a szintetikusan generált diszkrét maszkokhoz.

A 7. fejezetben a *nucleAIzer* [39] nevű módszerünket mutatjuk be. Az eljárás a Data Science Bowl 2018 verseny [14] keretében lett kifejlesztve sejtmagok példány alapú szegmentálásához heterogén képeken. A bemutatott eljárás egy előzőleg betanított régió alapú szegmentáló módszerrel (R-CNN) gyenge szegmentációkat hoz létre a teszthalmaz egyes képeihez. Mivel az adatbázis heterogén képekből áll, melyeket különböző technológiákkal hoztak létre (fénymikroszkóp, fluoreszcens mikroszkópkép, szöveti kép, stb), ezért az első lépés a teszthalmaz képeinek a klaszterezése. Ehhez a tanító adathalmaz elemeiből egyszerű jellemzőket nyerünk ki, majd egy egy rejtett réteggel rendelkező neurális hálózatot tanítunk be arra, hogy megállapítsa két kép között a hasonlóságot a jellemzők alapján. A betanított modellt alkalmazzuk a teszthalmaz képeire, majd a keletkező hasonlósági mátrix alapján azokat a k-means algoritmus segítségével klaszterezzük. Az egyes klasztereken a gyenge szegmentációkat tartalmazó maszkok és a hozzájuk tartozó képek alapján image-to-image translation [41] modelleket tanítunk, majd szintetikus maszkokat generálunk, és a betanított modellek segítségével létrehozzuk a szintetikus maszkokhoz tartozó szintetikus képeket. Az generált példákat hozzáadjuk a tanítóhalmazhoz, majd a modellt a kibővített adathalamzon tanítjuk. Ennek a módszernek az egyik gyenge pontja az, hogy a maszkokat egy egyszerű parametrikus módszerrel [53] állítottuk elő. Ez megfelelő lehet abban az esetben, ha sejtkultúrákat szeretnénk szimulálni, vagy olyan szöveteket, amelyekben a sejtek egyszerű eloszlással szimulálhatóak.

A 8. fejezetben prezentálunk két olyan szöveti képekből álló adatbázist, amelyben komplexebb struktúrák figyelhetők meg. Ezeken a képeken a sejtek elhelyezkedését nem lehet egyszerű parametrikus modellekkel reprezentálni. A módszerünk a maszkokon található objektumok eloszlásainak reprezentálására egy GAN [43] modellt használ, melyet a maszkokból származtatott hőeloszlás térképeken tanítunk be. Megmutattuk, hogy egy megfelelő kódolási technika elengedhetetlen ahhoz, hogy a GAN modelleket diszkrét maszkok szimulására lehessen használni. A mi módszerünkkel szintetikusan generált maszkok és a tanítópéldákban szereplő maszkok eloszlása szignifikánsan közelebb van egymáshoz, mintha azokat egy parametrikus módszer [53] segítségével hoztuk volna létre. A mikroszkópos képek generálásához ebben az esetben is az image-to-image translation modelleket tanítunk a tanító adathalmaz elemein, majd a szintetikusan generált maszkokra alkalmazzuk azokat. Megmutattuk, hogy

amennyiben a szintetikus képeket is felhasználjuk a tanításhoz, akkor jobb eredményt érünk el a teszthalmazon.

A tézispontok összefoglalása magyar nyelven

A tézispontok részletes kifejtése az 1. fejezet 1.3 szekciójában olvasható a disszertáció nyelvén. Jelen szekcióban a tézispontok tömör összefoglalása következik.

A munka három tézispontot tartalmaz. Ezek közül az első kettő a disszertáció I. részét fedi le, míg a harmdaik tézispont eredményei a disszertáció II. részében vannak bemutatva.

- Az első tézispontban egy gyors és pontos level set reinicializációs módszert javaslunk, amely a phase field elméleten alapul. Empirikusan megmutatjuk, hogy az eredeti phase field funkcionál a kontúrt elmozditja, noha level set reinicalizációként használva a feladata csupán az lenne, hogy a kontúr környezetében visszaállítsa a level set előjeles távolságfüggvény-tulajdonságát. A phase field funkcionál Euler-Lagrange egyenletéből látható, hogy ezt a hatást a Laplace összetevő okozza. A javasolt módszer az eredeti gradiens, és egy másodrendű tag megfelelő kombinációjával kompenzálja ezt a görbületfüggő mozgást. Empirikusan bebizonyítjuk, hogy a mi módszerünk jelentősen kisebb hatást fejt ki a kontúrra, mint a többi közelítő reincializációs módszer, és hasonló tulajdonságokkal bír, mint a pontos reinicializációs módszerek, ugyanakkor a számítási igénye sokkal kisebb. Az eredmények a 3. fejezetben vannak részletesen kifejtve.

- A második tézispontban egy 2D szelektív aktív kontúr módszert adaptálunk 3 dimenzióra. A szelektív aktív kontúr módszer a térfogatot, a felületet, illetve ezek kombinációjaként előálló alakleíró jellemzőket használja fel az aktív kontúr evolúció során. Bemutatjuk, hogy a javasolt priorok 3 dimenzióban is működnek. A kifejlesztett aktív kontúr módszert kombináljuk az első tézispontban javasolt reinicializációs módszerrel, és egy lokális régió alapú adattaggal. Az eredményként előálló algoritmust beépítjük egy 3 dimenziós orvosi képfeldolgozó szoftverbe (MITK). Az előálló módszerrel hasonló pontossággal lehet 3 dimenziós képeken objektumokat annotálni, mintha a kontúrokat 2D szeletenként rajzolnánk meg, ugyanakkor a mi általunk kidolgozott szoftver jelentősen lerövidíti az annotálási időt. Releváns fejezetek: 4, 5.

- A harmadik tézispontban két augmentációs eljárást dolgozunk ki. Az eljárások az image-to-image translation módszert használják arra, hogy szimulált objektummaszkokhoz rendeljenek hozzá mikroszkópos képeket. Megmutatjuk, hogy a szintetikusan generált mikroszkópos képek hozzáadása a tanító-adathalmazhoz

javítja a modell általánosító képességét. Az első módszer a DSB 2018 sejtmag detekciós adatbázisra lett kifejlesztve, melyben a maszkokat egy parametrikus sejtpopuláció szimulációs módszerrel állítjuk elő. Ennek azonban a hatékonysága limitált, ugyanis egyszerű parametrikus módszerekkel nagy kihívást jelent a komplexebb szöveti struktúrákban megfigyelhető objektumeloszlások szimulálása. Ezért egy további módszert mutatunk be, amely kiküszöböli az első módszer ezen limitációját, és egy GAN alapú eljárást javaslunk a maszkok szimulációjára. Releváns fejezetek: 7, 8.

A tézis szerzőjének hozzájárulásai a tézispontokban felsorolt eredményekhez

- Hozzájárulások az első tézisponthoz:
  - A javasolt phase field modellt [BPF] beágyaztam különböző aktív kontúr modellekbe (geodéziai aktív kontúr modell, szelektív aktív kontúr modell 2D és 3D megvalósítása), az egyenletek numerikus megoldása GPU-n és MATLAB-ban.
  - Az összehasonlított modellek implementálása:
    * egy fejlett, parciális differenciálegyenlet alapú reinicializációs módszer implementálása [RM] (Peng et al., 1999)
    * a Reaction-Diffusion reinicializációs módszer implementálása method [RD] (Zhang et al., 2013).
  - A javasolt reinicializációs módszer validálási folyamatának megtervezése és végrehajtása:
  - Mikroszkópos tesztek (valós adatok): egy geodéziai aktív kontúr modellbe ágyaztam be a javasolt phase field reinicializációs módszert. Kiválasztottam a DSB 2018 sejtdetekciós adathalmazból objektumokat (mindegyik kivágott kép egy sejtet tartalmaz), összehasonlítottam a BPF, a RD és az RM módszereket úgy, hogy kiszámítottam a maximális átfedést (IoU, intersection over union) amely az aktív kontúr evolúció során elérhető volt az egyes reinicializációs módszerekkel.
  - Létrehoztam szintetikus bináris maszkokat, amelyekkel inicializáltam a kontúrt. A sebességfüggvényt identikusan 0-ra állítottam, és a reinicializációs módszerek stabilitását teszteltem. A hipotézis szerint azon módszerek, amelyek mellékhatást fejtenek ki a kontúrra, elegendő számú iteráció végrehajtása után jelentősen megváltoztatják a kezdeti kontúr topológiáját, noha a sebesség mindenhol 0 (kis objektumok eltűnnek, közeli objektumok ”összeolvadnak”).

  - Reprodukáltam a RD cikkben található 3 dimenziós súlyzó alakú objektummal “dumbbell” végrehajtott kísérletet (3D aktív kontúr), mely egy görbület által hajtott folyamatot szimulál. Összehasonlítottam, hogy mi történik az egyes reinicializációs módszerek esetén. A teszt kimutatta, hogy ugyananakkora számú iteráció végrehajtása után a RD reinicializáció használatával a folyamat ”előrébb tart” a kezdeti állapothoz képest, mint a RM vagy a BPF esetében, az RD modellben található simasági tag mellékhatásaként.

- Hozzájárulások a második tézisponthoz:
  - A 2D szelektív aktív kontúr modellt (Molnar et al. 2016) adaptáltam 3 dimenzióra. A hatékony numerikus megoldás érdekében a módszert GPU-n implementáltam (kivéve a kezdeti implementációt, ami a marching cubes algoritmusból, a térfogat priorból, a simasági tag implementációjából, és egy egyszerű él-alapú adattagból, és OpenGL vizualizációból állt).
  - Az alak prior kifejesztésén dolgoztam (az objektum térfogatának és felületének a függvénye). A numerikus hibákat vizsgáltam, különböző reinicializációs módszereket teszteltem. Megmutattam, hogy az Euklideszi távolság alapján újrainicializált level set függvénnyel ezen numerikus hibák kiküszöbölhetőek, és a felület felveszi az alakleíró számmal meghatározott alakot. Továbbá megmutattam, hogy az első tézispontban javasolt phase field reinicializációs módszer használatával az alak prior hasonlóan megvalósítható.
  - Megvalósítottam a GPU implementációját különböző reinicializációs módszereknek:
    * Eikonal differenciálegyenlet,
    * Chamfer-távolság,
    * Euklideszi (egzakt) reinicializáció.
  - Teszteltem az alak priort a megvalósított reinicializációs módszerekkel.
  - Megvalósítottam egy algoritmust, amellyel GPU- kiszámítottam a level set által reprezentált kontúr összefüggő komponenseit, az objektumok kinyerésének a céljából.
  - Külön sebességfüggvényt alkalmaztam az egyes kinyert objektumokra, így megvalósítva a szelektív szegmentálást.
  - Megvalósítottam egy hatékony lokális level set algoritmust, amely csak a felület kis környezetében tartja karban a level set függvényt, ezáltal jelentősen csökkentettem azon pontok számát, ahol a sebességfüggvényt újra meg kell határozni.
  - Demonstráltam egy szintetikus képen, hogy a térfogat prior meg tudja különböztetni az objektumokat térfogat alapján (csak azok a komponensek maradnak meg, amelyeknek a térfogata megegyezik az előre meghatározott térfogattal).

  - Egy valós 3 dimenziós mikroszkópos képen mutattam meg, hogy a szelektív aktív kontúr modell működik. Élesztő sejtekről készíttettünk 3 dimenziós képeket egy nagy felbontású konfokális mikroszkóppal, kiválasztottam a régiót, amely két sejtet tartalmaz, az egyik élesztősejt gömb alakot vesz fel, míg közvetlenül mellette egy másik pedig hosszúkás formájú. Növeltem a kép felbontását a z-dimenzió (mélység) mentén egy optikai áramlás algoritmus alkalmazásával.
  - Kitaláltam és megterveztem egy 3 dimenziós képeket egysejt szinten annotáló folyamatot, melyből egy szoftvert hoztam létre.
  - A szoftverhez a szelektív aktív kontúr modellt párosítottam az első tézispontban javasolt phase field reinicializációs módszerrel, és egy fejlett, régió alapú adattaggal, melyet ugyancsak GPU-n valósítottam meg.
  - Létrehoztam a GPU-n futó kódból egy dinammikusan betölthető könyvtárat, és egy C++ interfészt.
  - Készítettem egy plugint az MITK orvosi képfeldolgozó szoftverhez, amelyben egy felhasználó iterfész segítségével lehet irányítani a szelektív aktív kontúr evolúciót a priorok, és az egyenlet egyes tagjainak a beállításával, mely a folyamat közben is dinamikusan változtatható. A szegmentált felület az MITK-ban van felületként renderelve, illetve a felület metszetei a háromnézeti paneleken is meg vannak jelenítve 2 dimenzióban.
  - Megvalósítottam egy adaptív algoritmust, amely az aktuális kontúr környezetét dinamikusan nyeri ki, és csupán a szükséges részleteken tartja karban a level set függvényt.

- Hozzájárulások a harmadik tézisponthoz:

  - A pix2pix image-to-image translation módszert alkalmaztam a Data Science Bowl 2018 verseny alatt ahhoz, hogy szintetikus tanítópéldákat hozzunk létre, amelyek a hipotézisünk szerint növelik majd a kezdeti modell általánosító képességét a teszt halmazra vonatkozóan.
  - Egy kezdeti szegmentáló hálóval ”gyenge” szegmentációkat generáltam a teszt halmaz elemeihez.
  - A teszthalmaz klaszterein pix2pix modelleket tanítottam, amelyek a maszkokat mikroszkópos képekké transzofmálják.
  - A szintetikus maszkokhoz létrehoztam a szintetikus mikroszkópos képeket az előzőleg betanított pix2pix modell segítségével.
  - Leimplementáltam a kiértékelési metrikát.
  - Részt vettem a `https://nucleaizer.org` kifejlesztésében, amelyen keresztül a felhasználók be tudják küldeni a képeiket, majd a modellünk

segítségével kiszegmentáljuk azokat, és elérhetővé tesszük az eredményeket a felhasználók számára.

- Készítettem egy szkriptet, amely a módszer minden elemét lefuttatja.
- Egy plugint készítettem a Napari képanalízis szoftverhez, amelyből lehet a modellt futtatni. `https://www.napari-hub.org/plugins/napari-nucleaizer`.
- Kidolgoztam a nucleAIzer módszer továbbfejlesztését, amelyben összetett struktúrájú szöveti képek diszkrét maszkjait szimulálhatjuk.
- A binarizált maszkok tanulása DCGAN segítségével.
- DCGAN tanítása a StarDist (Schmidt et al., 2019) módszerben leírt maszk-reprezentáció használatával.
- Kidolgoztam egy GVF (Gradient Vector Flow) alalpú kódoló-dekódoló módszert a diszkrét maszkok GAN-nal való reprezentálásához.
- A StyleGAN2-ada modellt tanítottam különböző maszk-reprezentációkon, melyekből bemutattam, hogy a Cellpose cikkben bemutatott hőeloszlás reprezentáció dekódolható a GAN-nal való generálás után.
- Kiválasztottam két adathalmazt, amely komplex szöveti struktúrákat tartalmaz (salivary gland tumor és fallopian tube adathalmazok).
- Létrehoztam szintetikus kéepeket a SIMCEP nevű parametrikus sejt-szimulációs módszerrel, hogy összehasonlítsam a generált maszkokon található objektumok eloszlását a StyleGAN által generált maszkokon található objektumok eloszlásával.
- Módosítottam a StyleGAN2-ada módszer kiértékelő kódját, amely két eloszlás közötti fID értékeket számítja ki az ImageNet által kinyert jellemzők alapján, hogy két adathalmaz is összehasonlítható legyen.
- Alkalmaztam a pix2pix modellt (amit az eredeti tanító-adathalamzon tanítottam be), hogy a szintetizált maszkokhoz hozzárendeljem a mikroszkópos képeket. A szintetizált képeket létrehoztam mind a SIMCEP, mind a StyleGAN által generált maszkokhoz.
- Megmutattam, hogy az fID távolság tekintetében a szintetizált maszkok közelebb vannak az eredeti adathalmaz maszkjaihoz, mint a SIMCEP módszerrel generált maszkok. Ugyanezt az eredményt kaptam a szintetizált mikroszkópos képekre is.
- Felosztottam az adathalmazt részekre, hogy keresztvalidációt végezhessek. Minden foldban az eredeti képekhez hozzáfűztem a foldon tanított modellekből szintetetizált példányokat (maszk, és mikroszkópos kép). Minden foldon isntance segmentation modelleket tanítottam (StarDist és Cellpose), az eredményeket kiértékeltem.

– Az eredeti tanító adathalmazból kiválasztottam részhalmazokat, és az azokon keresztvalidációval tanítottam StarDist és Cellpose modelleket mindkét adathalmazon. A kiértékelés megmutatta, hogy minden esetben javulás érhető el a teszthalmazon, amennyiben a szintetikus képeket is használjuk.
– Bemutattam, hogy a StyleGAN-ada modell effektíven tanítható transfer learning módszerrel. A salivary gland tumor adathalmazon inicializáltam a modellt, majd a fallopian tube adathalmazon tovább tanítottam. A modell már 1000kimg iteráció után is elfogadható pontosságot ér el az fID metrikát használva, ugyanezt elvégeztem a részhalmazokból kinyert maszkokon is.

The author's relevant publications

Journal publications

[1] **Ervin Tasnadi**, Alex Sliz-Nagy, Peter Horvath. Structure preserving adversarial generation of labeled training samples for single cell segmentation. *Cell Reports Methods*, IN PRESS, 2023.

[2] Reka Hollandi, Nikita Moshkov, Lassi Paavolainen, **Ervin Tasnadi**, Filippo Piccinini, Peter Horvath. Nucleus segmentation: towards automated solutions. *Trends in Cell Biology*, 32(4), 295-310, 2022.

[3] **Ervin A Tasnadi**, Timea Toth, Maria Kovacs, Akos Diosdi, Francesco Pampaloni, Jozsef Molnar, Filippo Piccinini, Peter Horvath. 3D-Cell-Annotator: an open-source active surface tool for single-cell segmentation in 3D microscopy images. *Bioinformatics*, 36(9), 2948-2949, 2020.

[4] Reka Hollandi, Abel Szkalisity, Timea Toth, **Ervin Tasnadi**, Csaba Molnar, Botond Mathe, Istvan Grexa, Jozsef Molnar, Arpad Balind, Mate Gorbe, Maria Kovacs, Ede Migh, Allen Goodman, Tamas Balassa, Krisztian Koos, Wenyu Wang, Juan Carlos Caicedo, Norbert Bara, Ferenc Kovacs, Lassi Paavolainen, Tivadar Danka, Andras Kriston, Anne Elizabeth Carpenter, Kevin Smith, Peter Horvath. nucleAIzer: A Parameter-free Deep Learning Framework for Nucleus Segmentation Using Image Style Transfer. *Cell Systems*, 10(5), 453-458, 2020.

[5] Filippo Piccinini, Tamas Balassa, Antonella Carbonaro, Akos Diosdi, Timea Toth, Nikita Moshkov, **Ervin A Tasnadi**, Peter Horvath. Software tools for 3D nuclei segmentation and quantitative analysis in multicellular aggregates. *Computational and structural biotechnology journal*, 18, 1287-1300, 2020.

[6] Ilaria De Santis, **Ervin Tasnadi**, Peter Horvath, Alessandro Bevilacqua, Filippo Piccinini. Open-Source Tools for Volume Estimation of 3D Multicellular Aggregates. *Applied Sciences*, 9(8), 1616, 2019.

Full papers in conference proceedings

[7] Jozsef Molnar & **Ervin Tasnadi**, Balint Kintses, Zoltan Farkas, Csaba Pal, Peter Horvath, Tivadar Danka. Active surfaces for selective object segmentation in 3D. In *2017 International Conference on Digital Image Computing: Techniques and Applications (DICTA)*, IEEE, 669-675, 2017.

[8] Jozsef Molnar, **Ervin Tasnadi**, Peter Horvath. A Balanced Phase Field Model for Active Contours. In *Scale Space and Variational Methods in Computer Vision*, Springer International Publishing, 419-431, 2019.

Acknowledgments

I would like to thank my supervisor, Peter Horvath, for directing my PhD studies. I would also like to thank Jozsef Molnar, Filippo Piccinini, Csaba Molnar, Krisztian Koos, Reka Hollandi and all of my collaborators who helped me to realize the results presented here.

Furthermore, I have to mention the complex exam committee: Csaba Benedek, Kalman Palagyi, Laszlo Czuni, Zoltan Kato and Tamas Vinko who gave useful advices for the second part of my PhD studies.