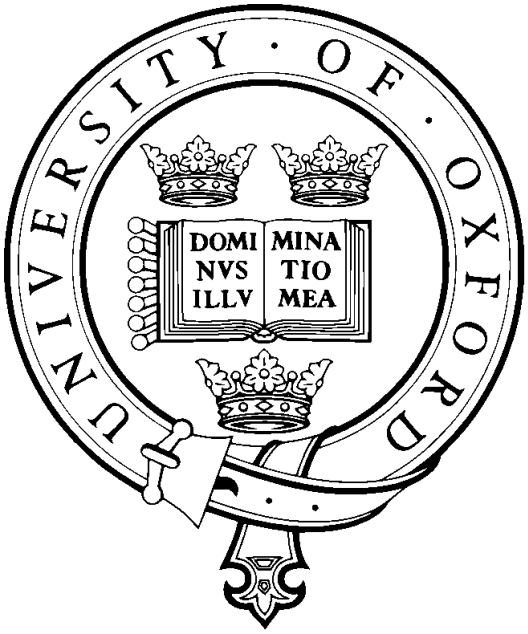
Computer vision and machine learning for microscopy image analysis



Carlos Arteta

Pembroke College

University of Oxford

Supervised by
Dr. Victor Lempitsky, Professor J. Alison Noble and Professor Andrew Zisserman
Submitted: Trinity Term 2015

This thesis is submitted to the Department of Engineering Science, University of Oxford, in fulfilment of the requirements for the degree of Doctor of Philosophy

Abstract

This thesis presents methods that address three fundamental tasks in the field of microscopy image analysis: detection of instances of an object, counting instances of an object when individual detection is not possible, and exploration of large datasets of microscopy images. We address these areas while strongly considering the constraints related to the usability and flexibility of a real practice. Through the use of computer vision and machine learning, we aim to deal with minimalistic user annotations, interactivity and intuitive data visualization.

Firstly, a novel framework is presented to detect all the instances of an object of interest in microscopy images, where they may be partially overlapping and clustered. To this end, a tree-structured discrete graphical model is introduced, that is used to select and label a set of non-overlapping candidate regions in the microscopy image by a global optimization of a classification score. Additionally, it is shown how to learn, from the images, the generation of a surface that can improve the collection of the candidate regions. The learning for the object model and optimal surface only require simplistic annotations – a dot on each instance of the object.

Secondly, it is considered the scenario where individual instances of an object may not be discernible due to extreme overlap, but the count of instances over regions of interest is still useful quantitative information. A novel object density estimation approach is proposed, which is also trained from simple dot-annotations, and achieves a similar accuracy to current state-of-the-art methods while being considerably faster to train. The application to counting in time-lapse microscopy sequences is illustrated. Moreover, by taking advantage of the possibility to learn on-the-fly from simple annotations, an interactive counting system is developed which allows a user to quickly obtain object counts on simple cases, or to bootstrap dataset annotations for more complex scenarios.

Finally, the exploration of microscopy datasets coming from large exploratory studies is considered. An unsupervised pipeline is proposed, which allows the discovery and visualization of the effect of external perturbations such as drugs over a target of interest (e.g. a cell protein). This framework enables a user to perform large scale image-dataset exploration in an intuitive way using a novel 2D visualization tool.

Acknowledgements

I am extremely grateful to my wonderful group of supervisors, Prof. Alison Noble, Dr. Victor Lempitsky and Prof. Andrew Zisserman, whose constant guidance, enthusiasm and infinite patience enabled the completion of this thesis. A big thank you to the VGG members for making the lab an excellent environment. I am also grateful to Prof. Fred Hamprecht and Prof. Jens Rittscher, not only for reviewing this thesis, but also for the useful discussions and encouragement during my DPhil. Finally, all my appreciation to my fiancée and family for the many years of unconditional support.

Contributors and collaborators Parts of this thesis would have not been possible without a series of close collaborators and contributors, acknowledged next.

The application of cell counting for lens-free microscopy of Section 6.4 was done in close collaboration with researchers at the Gray Institute for Radiation Oncology and Biology, University of Oxford, including Dr. Giselle Flacavento, Dr. Boris Vojnovic, Dr. Paul Barber, Dr. James Thompson and Dr. Lisa Folkes.

The data exploration method of Chapter 8 was developed in close collaboration with Jaroslav Zak and Dr. Xin Lu from the Ludwig Institute for Cancer Research, University of Oxford, who were also supported by the microscopy facilities and research staff at the Target Discovery Institute, University of Oxford.

Finally, several of the datasets used in this thesis for the task of object detection in microscopy images (Chapters 3-5) were kindly provided as follows. The researchers of the Laboratory for Viral RNA Biochemistry, Institute of Protein Research RAS, provided the images and annotations of gels with molecular colonies; Dr. Svetlana Uzbekova (INRA, Physiology of Reproduction and Behavior Unit, Nouzilly, France) provided the images of fluorescent nuclear stained bovine blastocysts; Dr. Boris Vojnovic and James Thompson (Grey Institute for Radiation Oncology and Biology, University of Oxford, UK) provided the equipment and samples for the collection of the phase contrast dataset; Dr. Nasir Rajpoot provided the histopathology dataset; and Dr. Julian Gingold provided the dataset of cell nuclei in fluorescence microscopy.

Contents

Chapter 1: Introduction

1.1 Motivation

Modern automated microscopy platforms have the capacity to generate massive amounts of experimental data at a high pace. Such systems have been widely adopted over the past decade both in academia and industry, resulting in a crucial need for methods that can also accelerate and improve the analysis of the large microscopy-based experiments.

It has been shown that, in combination with the appropriate imaging platforms and labelling techniques, image analysis methods can automatically provide the quantitative information that is required to better understand complex biological systems [124]. The areas of application are numerous; for example, drug discovery, tissue engineering, genomics and proteomics are some of the sciences that greatly benefit, and to some extent, depend on accurate, fast and intelligent cellular image analysis algorithms.

Techniques for solving the fundamental problems within large-scale microscopy image analysis have then emerged, but their adoption is by no means straightforward. As it happens in the general biomedical image analysis field, the transition from successful development of methods into their adoption by the end users can be slow or non-existent, with the exception of the cases where the developers and users work closely together. The adoption problem exists partially due to the highly interdisciplinary nature of the field. For example, developers might not fully understand the needs or work-flow of the end-users (e.g. biologists), meanwhile end-users might not understand the virtues or limitations of certain methodology, or how to apply it in their specific scenario. In order to alleviate this problem, there are many requirements that microscopy-image analysis (and

biomedical imaging) software must fulfill, as highlighted by Carpenter *et al.* [28]. Fortunately, thanks to the great effort of interdisciplinary research groups, open-source software packages have been developed which facilitate the transition of modern microscopy-image analysis methods into the practice of early adopters; some examples are FIJI [133], Ilastik [146] and CellProfiler [27], as well as the web-based microscopy-image data management system OMERO [3]. With the existence of such packages, an important proportion of the burden for the dissemination and usage of microscopy-image analysis methods is reduced, leaving the developers with the tasks intrinsic to the method development. Among those, a key task is that of user-friendlessness.

For a method to be easy to use while maintaining the accuracy required for the application, the ease of use objective must often be a design consideration of the method itself, and not only a software package design problem. For example, if a method requires user input in order to adapt to a new scenario, it must consider how to handle the easiest type of user input it could, which can translate into how to use labels that are intuitive for the user, of a minimalistic type, prone to errors, or how to operate while requesting the minimum amount of input possible.

A natural way to develop methods with the flexibility required to adapt to unseen scenarios with the help of the end user is through machine learning techniques. Learning-based methods, in combination with computer vision, have the capability of learning from the end users how to perform a specific task in images while only requesting inputs related to their area of expertise; all these while being accurate and reliable for when quantitative information is required. Therefore, it is expected that the role of machine learning within the microscopy-image analysis field will continue to grow and become indispensable [110] for many biology-based sciences.

This thesis develops methods for microscopy-image analysis that address various of the fundamental tasks in the field while strongly considering the constraints related to the usability and flexibility in a real practice. In more detail, throughout the methods proposed, we aim to deal with minimalistic user annotations, interactivity and intuitive

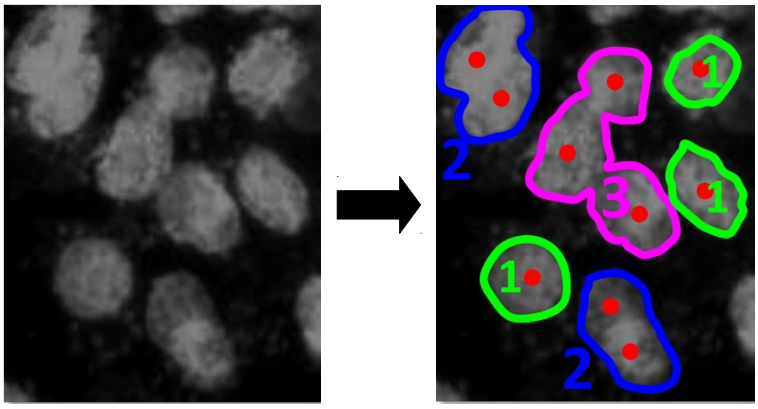


Figure 1.1: (Object detection example) *Given an input image containing multiple instances of an object in a microscopy image, we aim to detect them all even when these may be overlapping.*

data visualization. The specific contributions are described next.

1.2 Thesis outline and contributions

The areas addressed within this thesis can be clustered into three categories: (i) detection of objects in microscopy images, (ii) estimation of object density, and (iii) exploration of large microscopy datasets, which are presented in Chapters 3-8, as detailed below. Additionally, a review of work related to the three areas of research is presented in Chapter 2, and finally, an overall conclusion and future work is presented in Chapter 9.4.

1.2.1 Detecting objects in microscopy images

Automatic detection of objects in microscopy images is a subject of interest in a wide range of experimental procedures. A common example is cell detection (see Figure 1.1), which can be fundamental for cell-based studies as it is the starting point of many automatic methods for cell counting, segmentation and tracking. The broad diversity of factors such as cell lines, dish confluency, and microscopy imaging techniques require that cell detection algorithms adapt well to different scenarios. Moreover, in some applications, additional cell types or other similar structures can be present in the same images, requiring the algorithm to be able to discriminate and detect only the phenotypes of interest, which poses a barrier hard to overcome with classical image processing

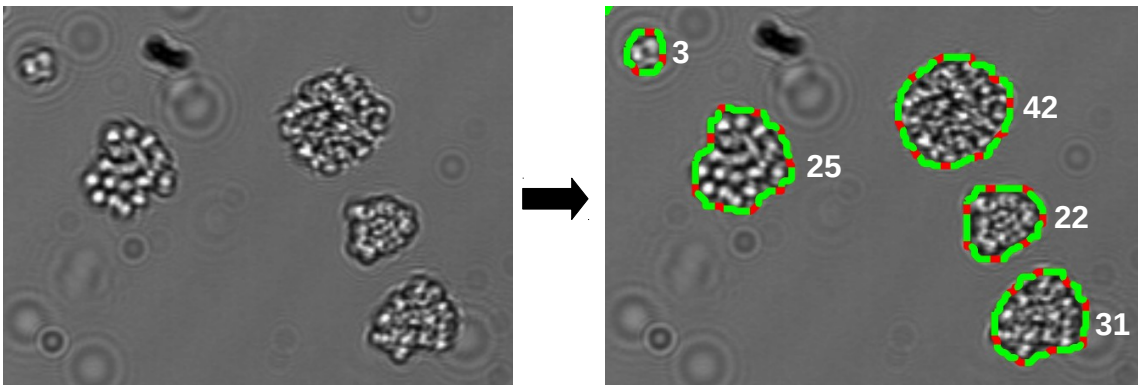


Figure 1.2: (Object counting example) *Given an input image containing multiple instances of an object in a microscopy image, we aim to count them all even when the individual instances are not discernible and cannot be detected.*

techniques.

A method with such adaptability can greatly benefit from detailed user annotations to learn a model of the objects of interest. Nevertheless, the labour required for extensive user input such as pixel-wise annotations diminishes the cost-benefit of the method. Therefore, we aim to achieve the detection task with relatively simple expert supervision: dot annotations.

The contributions towards object detection in microscopy images are distributed as follows. Chapter 3 presents a method that is able to learn a discriminative model of an object of interest in microscopy images from simplistic user dot-annotations. Such model is then used to detect all of the instances of the object in previously unseen images. This object detection model is then extended in Chapter 4 mainly for the purpose of handling the case where instances of the object of interest appear overlapping, as it is common, for example, in cell cultures of high confluency. Finally, Chapter 5 presents a pre-processing step which also uses the dot-annotations of a training set in order to learn a model that is used to enhance microscopy images in such a way that it facilitates object detection (i.e. with the detection method of the previous chapters).

1.2.2 Estimating object densities in microscopy images

As shown in the object detection chapters, it is possible to handle the detection of overlapping instances of an object in microscopy images with a good accuracy. Nevertheless, there are extreme cases where explicit object detection is not at all possible or where individual object detectors do not work reliably due to crowding, severe overlap, or reduced size of the instances. However, the explicit object detection might not be critical for the end goal. For example, one might be interested in studying how the proliferation rate in a cell culture is affected by some perturbagen (e.g. radiation or chemical), for which a useful piece of information would be the number of cells as a function of time (i.e. cell count; see Figure 1.2).

A natural way to count objects in images is by detecting them individually, but if this is not possible, an alternative is to learn a mapping from image features to a pixel-wise object density such that the object count over regions of interest can be recovered by simply integrating over the object density map. This alternative method is referred to as counting through object density estimation, and in some scenarios, can even allow the localization of the objects of interest. The approach of density estimation is described in Chapter 6, where we contribute with a new and simple method to learn the mapping from local image features to object density, as well as a general approach to impose temporal smoothness for the scenario of counting on time-lapse sequences. Similar to the detection methods proposed in Chapters 3-5, the density estimation frameworks, such as those presented in this thesis, learn the density mapping from simple dot-annotations placed by the user on a training set.

As expected, when trying a supervised counting method on a new dataset, the user would wonder how much annotation is required. Depending on the difficulty of the problem (e.g. the complexity of simplicity of the object or background appearance), the number of annotations required to produce useful results can greatly vary. By using interactivity, we contribute in Chapter 7 with a counting method that allows the user to

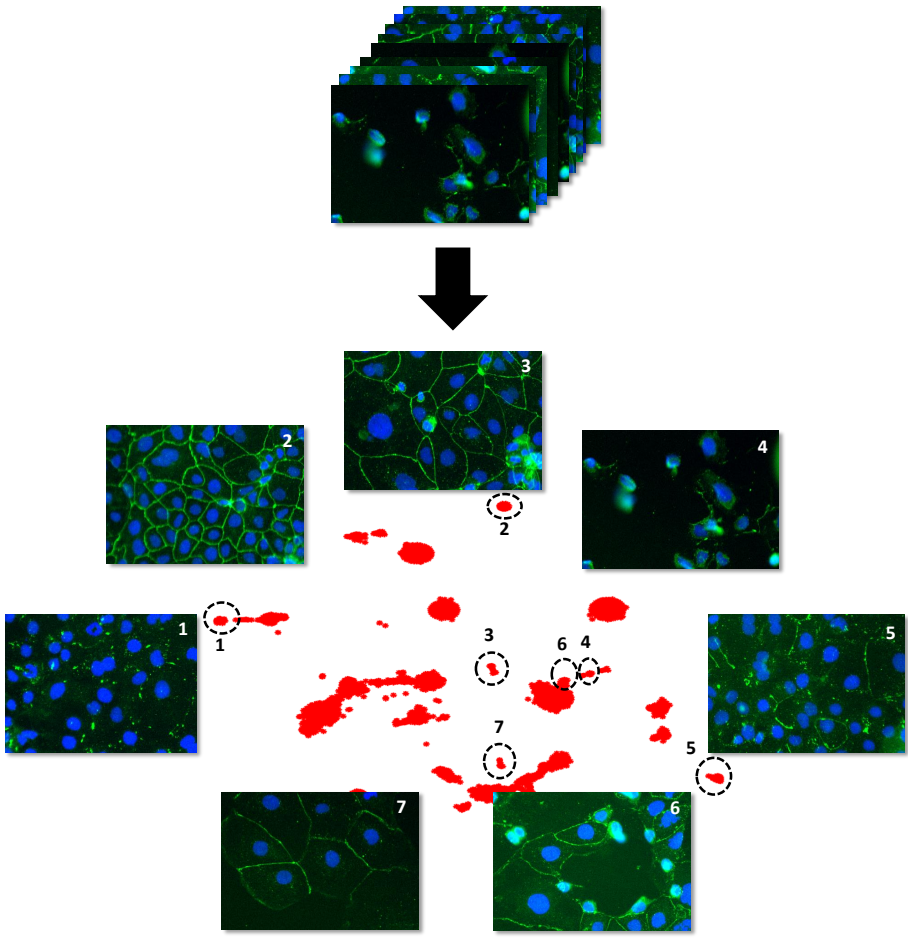


Figure 1.3: (Microscopy dataset exploration example) *Given a large dataset of microscopy images, we aim to produce intuitive visualizations that allow the easy exploration of the visual patterns in the data.*

assess in a matter of seconds the quality of the counting results given only a few annotations, and to update the results according to the accuracy demanded by the application. Once the user is satisfied, the density estimation function learned can be used to count objects in unseen images of a similar experimental setup. For the interactive to be efficient, it must be intuitive for the user which tasks need to be performed. Therefore, we also contribute with two object density visualization techniques that allow the user to quickly assess where the density estimation is failing the most in order to provide only the required annotations.

1.2.3 Exploring microscopy datasets

In the final area of this thesis, we address the problem of exploring large datasets of microscopy images, particularly applicable for the case of high-through screenings in exploratory studies (see Figure 1.3).

In fields such as drug discovery, a key step in the pipeline is that of identifying the possible effects over a target of interest (e.g. the translocation of a target protein). Nevertheless, the experimental setup for discovering such unknown effects often consists of very large screenings of preturbagens (e.g. small molecules in a common case of drug discovery), where thousands or millions of different combinations of experimental conditions are explored. For instance, the experimentalists might be interesting in exploring the effect of thousands of different compounds, over different cell lines, at different concentrations, and drug exposure times, among other possible variables, which can be performed with relative ease by using automated imaging platforms. The result of such a screening is datasets of potentially millions of images, which needs to be explored in order to assess what relevant effects can be caused on the target and which experimental variables caused them.

In Chapter 8 we propose a pipeline that uses whole-image (or whole-sample) characterization in oder to determine the similarities between the samples in the screening. By using these pairwise similarities, the method generates an intuitive visualization of the entire dataset where the different groups of visual patterns emerge. Furthermore, we show how this dataset visualization can be used to generate hypotheses of whether an experimental variable is involved in the generation of a visual pattern of interest.

1.3 Related publications

The work on object detection of Chapter 3 was initially published in MICCAI 2012 [9], followed by its extension presented in Chapter 4, which was published in CVPR 2013 [10]. Finally, the preprocessing method of Chapter 5 was published along with an extended

evaluation of Chapters 3 and 4 in the Journal of Medical Image Analysis 2015 [12].

The ridge-regression based density estimation method of Chapter 6 was published along with the interactive counting system of Chapter 7 in ECCV 2014 [11]. Application papers of counting on time-lapse sequences applied to lens-free microscopy and material formation are under preparation.

Finally, the publication of the methodology presented in Chapter 8 is also under preparation.

Chapter 2: Background

In this chapter we review the existing relevant literature that serves as background for the work presented in this thesis. First, we review the work that addresses three of the main challenges in automated analysis of microscopy images: object[1] detection, segmentation and counting, which closely relate to the research presented in Chapters 3 to 7. Secondly, we review the literature related to the exploration and visualization of large image-based screenings such as high-throughput screenings, closely related to the work presented in Chapter 8.

The structure of the review is the following: Section 2.1 covers different approaches for object detection and segmentation, Section 2.2 object description; Section 2.3 deals with the object counting methods, with emphasis on density estimation for counting objects in microscopy images; and Section 2.4 covers the aspects of characterization of entire microscopy images, as well as the exploration of datasets based on such characterization.

2.1 Detection and segmentation

Even though detection and segmentation are quite distinct tasks in the traditional computer vision literature, these tend to come closer together within microscopy image analysis. One possible reason is that a long-standing objective of detecting objects in microscopy images is that of segmenting them, but additionally, the much simpler appearance of objects of interest in microscopy (e.g. cells) allow methods to perform detection by

[1]Even though most of the literature reviewed is applied to cells in microscopy images, we use the word "object" to maintain the proper generality of the approaches, as these can be often applied to other organisms (e.g. bacteria) or even non-living objects (e.g. crystals) within the microscopy domain.

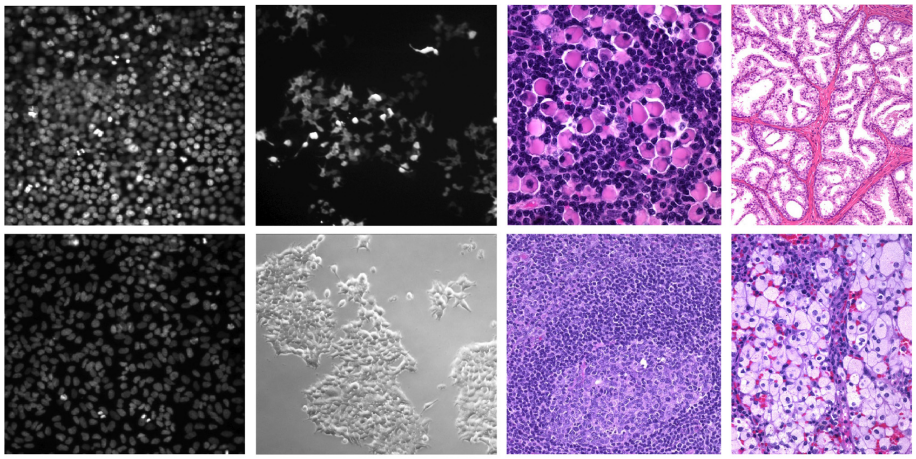


Figure 2.1: *Example of cell segmentation challenges [20]. Column 1: Epithelial A549 and embryonic kidney HEK293T cells featuring faint boundaries, varying dot intensities and occlusions. Column 2: More HEK cells of different convex geometries often appearing as cell clusters. Column 3: Histopathology images depicting tumour-like lesions: cells of various sizes and textures in spleen cell lymphoma. Column 4: spleen tumour cells metastatic melanoma, clusters of glands in ovarian cells and clear cell cribriform hyperplasia from prostate illustrating cells on non-homogeneous backgrounds.*

segmentation or *vice versa*, thus producing literature that mixes both tasks. Therefore, we review them jointly. Some of the challenges common to the detection and segmentation task include fading boundaries, low signal-to-noise ratios, presence of image artifacts (usually dependent on the microscopy modality), occlusion, presence of objects with similar appearance to those of interest, as well as shape, size and intensity variability of the objects within the same images (see Figure 2.1). The different strategies found in literature to solve the cell detection/segmentation can be grouped in four categories: 1) methods based on intensity thresholding and seeded watersheds; 2) methods based on active-contours; 3) methods based on blob detectors or other techniques to find local minima or maxima; and 4) methods based on statistical modelling or machine learning. Each category is described in more depth next.

2.1.1 Thresholding and watersheds-based algorithms

Approaches based on automatic thresholding algorithms, often combined with watersheds (or variations of it), are part of the initial attempts to perform automatic cell segmentation. However, they are still widely used in practice due to their simplicity, mainly

applied to simple cases of fluorescence microscopy, where the objects of interest are expected to show a higher intensity than the rest of the image. Watersheds have been used to separate cell clusters, but often lead to over-segmented images, thus shifting efforts towards region merging algorithms [91]. An early example of region merging is the method by Malpica *et al.* [98], which deals with segmentation of cell clusters in fluorescence microscopy images by doing an initial detection of cell nuclei through the h-dome transform [145], and then applying the watershed algorithm using the detected nuclei as seeds. Similarly, Lee and Street [87] used seeded watersheds to segment cells in 2D and 3D images. The seeds are generated applying the h-maxima transform [145] to the image of gradient magnitudes. The seeding is sufficiently sensitive to capture all cells, and the over segmentation is corrected by merging regions based on the gradient magnitude along the boundary separating neighbouring objects. Finally, a distance transform is used to separate cells in clusters that could not be resolved by means of edge detection. In the same way, Lin *et al.* [93] obtained over segmented cells through seeded watersheds based on adaptative thresholding, but the region merging is done by finding the best merging options on an adjacency graph. The nodes in the graph represent the regions out of the watershed segmentation, and the links connect nodes that correspond to neighbouring regions. All possible merges are scored considering how well the result of the merge agrees with a model of the cells based on size, shape, and convexity, and a recursive algorithm maximizes the overall score. Li *et al.* [91] proposed a 3D cell segmentation method that generates a smooth gradient field through a gradient vector diffusion and tracking procedure, and merges sink points based on distance between them. Then, adaptive intensity thresholding is done from each cell centre to segment the cell nuclei, managing to differentiate between cells is close proximity. Inspired by the theory of ensemble of weak classifiers, popular in machine learning, Debeir *et al.* [46] proposed an approach different to the common region merging used to find seeds for watersheds. A set of "weak" watershed transforms are applied to the image starting at seeds with added noise and under different image perturbations (e.g. brightness varied with a slope of random orientation).

The different results are added creating a map of agreement that can be thresholded. More recently, Ali *et al.* [2] explored how to use the visual information at different levels of focus in the microscope, and applied adaptive thresholding on images that result from the differences of images acquired at different focal planes, which can resolve cells in contact.

Even though the methods described perform well in the experimental set-up where they were tested, they generally require a significant amount of heuristics in the post-processing, posing a problem for the generalization and adaptation to changes in the datasets.

2.1.2 Active contours-based algorithms

Active contours, such as "snakes" [78] (extended in [106] to handle topological changes) or level-sets [136], have been widely used in segmentation tasks of all kinds within biomedical applications, and cell segmentation is a common example. The reason is not only that the active contours allow the effective use of intensity gradient or region information with imposed constraints (e.g. smooth shapes), but active contours can be readily extended to a tracking algorithm through contour evolution from frame to frame. Zimmer *et al.* [175] used parametric active contours based on [167] that can deal with low-contrast boundaries (e.g. fading boundaries or cell clumping) when combined with an edge enhancing process based on local average intensity deviation. However, the contours needed manual initialization. Mukherjee *et al.* [108] proposed a system based on level-sets for integrated cell detection and tracking that does not require manual initialization. Cell detection is done by minimizing, in a level-set framework, an energy functional that imposes shape (smooth elliptical shape), intensity (homogeneous region) and size constraints. Different priors have been proposed in cell segmentation within a level-set framework, such as Gaussian-shaped distributions [60], or shape models learned from the data [35]. A common problem when using level-sets for segmentation with the intention of doing contour evolution for tracking, is that region merging needs to be prevented for the cases where

cells are in close proximity. To do this, Dufour *et al.* [51] used one level-set per object along with a penalty for contour overlap. Attempting to eliminate the necessity of a unique level-set per object due to high computational cost, Nath *et al.* [111] proposed an algorithm for cell segmentation that uses at most four level sets with coupling constraints; adjacent cells are segmented with different level sets based on the "Four Color Theorem" [125], which states that any planar graph can be colored with at most four colors such that no neighboring vertices are assigned the same colour.

Another main issue in contour evolution algorithms is weak or missing edges. To address this, and following [175], Wang *et al.*, [163] used coupled snakes, but modified the energy function to include texture information that allowed the contours to be more sensitive to boundaries of the cell membrane rather than stronger edges inside the cell; this way, the contours could be initialized automatically with nuclei detection algorithms. Chen *et al.* [35] used subjective surfaces [132] to cope with missing edges under a shape constraint, and Ali *et al.* [2] used the monogenic signal [52] to extract local energy, phase and orientation, to guide the evolution of a level-set based active contour. On the other hand, Srinivasa *et al.* [147] proposed active masks, a change in the idea of cell segmentation based on active-contours where the evolution of the region is no longer based only on properties around the contour, but it takes into account the entire region that the contour is segmenting.

Contour initialization remains a critical point for active contours-based cell segmentation algorithms, since using the active contours also for cell detection is not flexible enough to accurately detect cells in complex cases, thus a separate detection algorithm is required.

2.1.3 Region detector-based algorithms

It is often convenient to model objects in microscopy images as regions of local minima or maxima, thus common blob detectors (or custom algorithms) have been applied for the microscopy object detection task. A popular blob detector for such a task is the

Laplacian-of-Gaussian [1,119,144]. Al-Kofahi *et al.* [1] detected cell nuclei with multiscale Laplacian-of-Gaussian filtering after a background removal based on graph-cuts. Cell clusters were divided with an additional step of graph-cut optimization based on alpha-expansion [24]. Bernardis and Yu [20] developed a sensitive method based on spectral graph partitioning to extract small convex regions from images, and demonstrated its application to cell detection, as well as detection of other small structures in biomedical images. Kuse *et al.* [84] detected cells in histopathology as regions with peak local isotropic phase symmetry, under the assumption that cells can be modelled as elliptical blobs; however, this is often not a valid assumption. Wienert *et al.* [165] used a grayscale closed contour detector to find cell candidates. The regions they find can be overlapping, which they solve by choosing the region with the highest strength of the intensity gradient along the contour.

Blob detectors do not have the flexibility to capture complex cell models. However, they can produce useful sets of candidate regions that can be further evaluated with more complex statistical models. An example of such use of blob detectors can be found within the detection method presented in Chapter 3, where maximally stable extremal regions $MSER$ [104] are used.

2.1.4 Learning-based algorithms

Data-driven statistical models, i.e. models learned from the data, in the task of object detection in microscopy images has been commonly used in two different ways: as a post-processing step, where hypotheses typically coming from the other types of detection methods are evaluated with supervised or unsupervised learning methods, or to generate hypotheses by classifying each pixel individually, followed by merging algorithms. Due to their flexibility (i.e. capacity to adapt to different microscopy scenarios) and reduced need for manual parameter tuning, learning-based methods have become the most popular approach for the tasks of detection and segmentation in microscopy. The detection method detailed in Chapters 3 and 4 fall within this category of detection methods, but

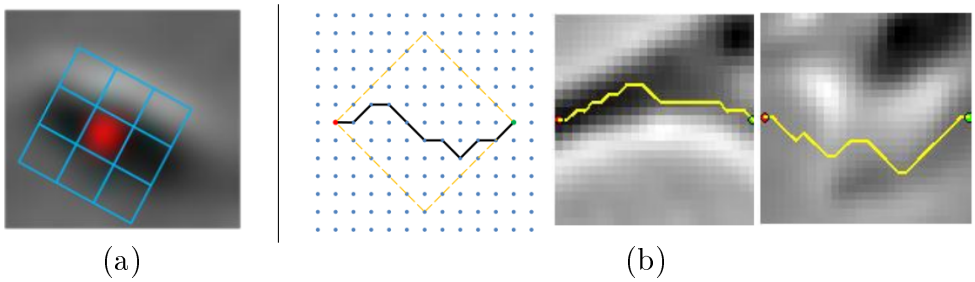


Figure 2.2: *(a) Illustration of the image block and sub-blocks from which features are extracted in [115], and (b) illustration of the search of the optimal connecting path between cell hypotheses. The dashed orange diamond delineates the search region, and a possible connecting path is illustrated with the black curve; examples are also shown in real images.*

with additional emphasis on using minimalistic annotations.

Many different learning techniques have been adapted for such a task; a few examples are described next:

Nattkemper *et al.* [112] used a neural network to compute the probability of each pixel in the image to be part of a cell, and all local maxima within distances of approximately the cell size are chosen as cell centroids. To train the neural network, image patches containing full cells are manually extracted from the image and marked as positive samples, whereas negative samples are randomly selected at a certain distance from the positive ones, thus possibly containing parts of cells. Each sample patch is encoded with the eigenvectors extracted from a principal component analysis of the sample patches, in the style of the "eigenfaces" [154]. After the cell centroids are detected, a square patch centred at each centroid is evaluated with a different neural network, which considers the dynamics of the gradients and the class probability of each pixel in the patch to produce a binary segmentation and delineate the closed contour containing the entire cell. Mao *et al.* [100] obtained over-segmented regions by thresholding on a grayscale image obtained from RGB images through max-margin class separability. The merging procedure was based on classifying the valleys between adjacent regions with a support vector machine. Similarly, Kong *et al.* [81] applied the same colour to grayscale conversion method, and used an SVM classifier to separate cells from background based on a Local Fourier Transform texture descriptor calculated over image patches. After the cell-

background separation, single cells and touching cells are identified with morphological features, and touching cells are split by breaking the clusters through concave points on the region boundary. Wei *et al.* [164] used an SVM classifier for cell segmentation in the application of cell counting through detection in *in-situ* microscopy. The training set consisted of fixed-size image patches containing whole cells, and the raw intensities were used as the feature vector. Multi-class classification was used, to firstly, segment cells from the background, and secondly, to differentiate between live and dead cells. Khairy *et al.* [79] proposed the use of Bayesian inference, together with a data-driven Markov-Chain Monte Carlo sampling technique, to fit a model of spatial organization, bending energy of labelled objects and topology, in order to segment cells in 3D fluorescence images. Pan *et al.* [115] used a combination of learning methods to detect cells in phase contrast images. After a rough background removal, candidate points are chosen as local minima of image intensity within a small search window to achieve high recall. Then, features including intensity information, intensity gradient (Laplacian filter) and HoG [45] are collected from divisions of a square block centred at each hypothesis (Figure 2.2(a)) and evaluated with a vector learned from a Gaussian-Kernel SVM. The scores obtained from the classifier are transformed to posterior probabilities using logistic regression, and thresholded to obtain a subset points that are within cells. Multiple points can still be within the same cell, thus a grouping algorithm is applied. The grouping is done by analysing the paths between small groups of detected points with dynamic programming, and comparing them with paths learned in the training (Figure 2.2(b)). Similarly, Marcuzzo *et al.* [102] used an SVM classifier to select cell regions out of a those produced by a watershed-based over-segmentation, and the region merging decision is done based on the strength of the intensity edges between candidate points. Cheng *et al.* [38] proposed a learning approach to select regions of interest in histopathology images out of a series of hypotheses generated with multi-phase level sets. Each candidate region was encoded with Zernike, Daubechies, wavelets, Gabor, skeleton, Haralick and morphological features, and Recursive-Feature-Elimination SVM [109] was used for feature selection.

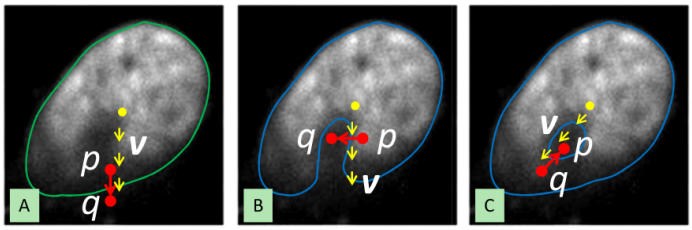


Figure 2.3: *Shape prior for cell segmentation in [95] by aligning cuts and some vector field. (a) Preferred segmentation, well aligned; (b) "crescent" shaped, orthogonal, and (c) a hole inside, opposite.*

Panagiotakis *et al.* [116] used a Mixture of Gaussian to model each different cell type in histopathology images. Using this model, regions of interest are detected and then discriminated with a transferable belief model using intensity, eccentricity, variance and size features. Yin *et al.* [168] performed pixel-level classification using a bag of local Bayesian classifiers, each trained on a different type of region found in the image through clustering based on intensity histogram. The weighting between the classifiers at test time depends on the histogram similarity of the patch being evaluated and the histogram of the cluster where each classifier was trained. Wu and Shah [166] proposed a conditional random field model to segment cells in multispectral images that models relations between objects in absorption images at different wavelengths. The energy functional has a local term, based on local information from low-level features and coarse information from latent topics discovery (using bag-of-words), a pairwise term with information of the edges between neighbouring patches, and a spectral term similar to the pairwise one, but connecting spectral neighbouring patches (third dimension). The CRF model is optimized with the graph-based method in [148]. Lou *et al.* [95] proposed a method for nuclei detection that uses a shape prior within a binary optimization based on graph cuts, where the weights of the energy terms are learned in a structured output framework. The shape prior consists of the alignment between the vector normal to the proposed cut (detected boundary) and the direction of the vector field (Figure 2.3), obtained from an euclidean distance map starting at cell nuclei. The shape prior favours blob-like structures.

These learning-based methods for object detection and segmentation can perform well

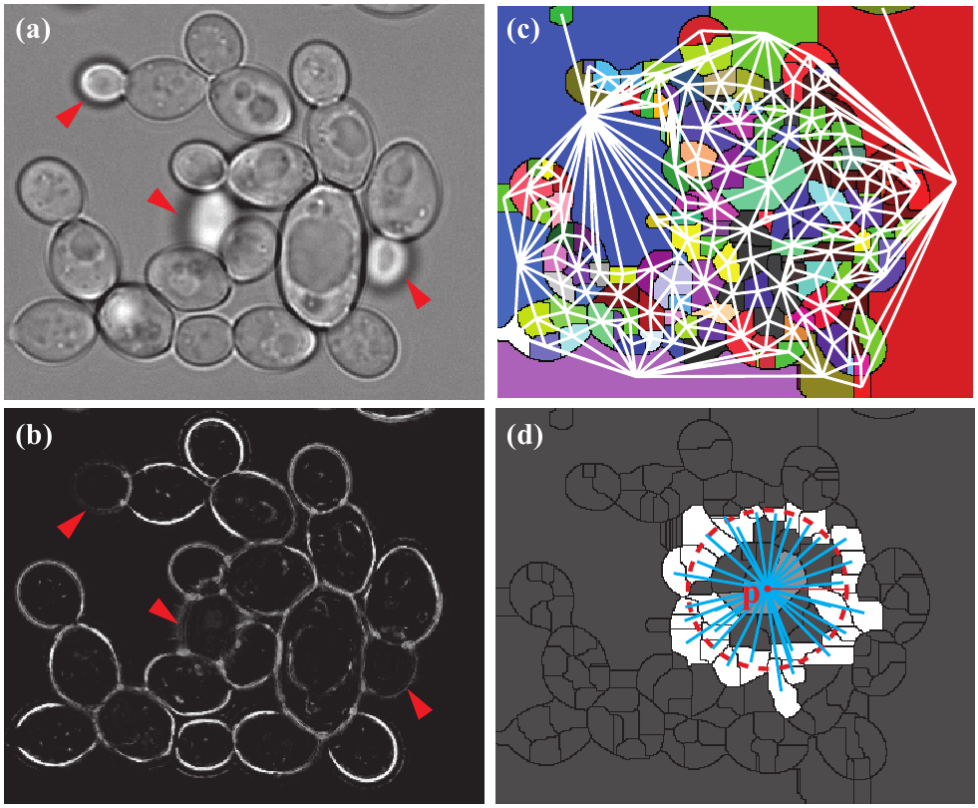


Figure 2.4: *Learning-based simultaneous object detection and segmentation method of Zhang* et al. *[170]. (a) On an unseen input image, the method can estimate (b) the probability of each pixel to be an object boundary. Then, (c) superpixels are collected from the boundary map, and used to build an adjacency graph where edge weights (in white) are influenced by the boundary probabilities. Finally, (d) a distance constraint is applied to discard edges between far-away superpixels before a correlation clustering procedure segments the graph, delineating the objects of interest.*

under different scenarios, but often require tedious annotations such as an extensive set of "positive" and "negative" patches. However, those problems can be overcome with modern learning techniques. A recent example is the simultaneous detection and segmentation method from Zhang *et al.* [170], which learns to estimated the probability of cell boundaries from a few boundary-annotations, which are then used as edge weights on a clustering of superpixels based on correlation clustering (see Figure 2.4).

2.2 Description

The description (and classification) of cellular organization and structure, such as nuclear architecture (e.g. localization of genes and proteins), staining patterns, fluorescence

intensity, and cell morphology, is a task of producing visual features that are discriminative enough to understand and quantify the difference between different populations. Moreover, the different description methods used to characterize cells in microscopy images can also serve as reference of hand-crafted features that can be collected to use in learning-based methods for several tasks within microscopy image analysis, such as detection, segmentation, tracking and sample characterization. Therefore, we briefly review the literature of two important areas of cell description in images, which serve as inspiration for the visual features used in the detection and counting methods described in Chapters 3-7.

2.2.1 Shape analysis

The analysis of cell shape, or cell morphometry, focuses on capturing biologically relevant shape variations between cell classes, e.g. under different experimental conditions [122]. Capturing shape variations is a common task in biomedical applications, however, opposite to macroscopic structures (e.g. organs), small objects in microscopy are generally not suitable to apply landmark-based registration methods, specially in label-free microscopy, thus methods encoding the entire cell shape are preferred. Pincus and Theriot [122] reviewed quantitative methods for cell-shape analysis, covering cell shape representation and encoding. Common cell shape representations are binary masks, feature-based (i.e. length of the principal axis), quantized curves and distance maps (i.e. distance from each pixel to the cell edge). Given the shape representation, encoding methods have been used to reduce the shape representation for compactness, but without losing much fidelity. For this purpose, Fourier decomposition, Zernike polynomials, principal components analysis (PCA), linear discriminant analysis (LDA) and independent component analysis (ICA) were compared in [122]. The cell shape analysis methods studied in [122] were evaluated in terms of representation fidelity, discriminative capacity and interpretability of the results, concluding that PCA of shape outlines provided the best overall performance. Rohde *et al.* [126] proposed a non-linear method based on large deformation diffeomor-

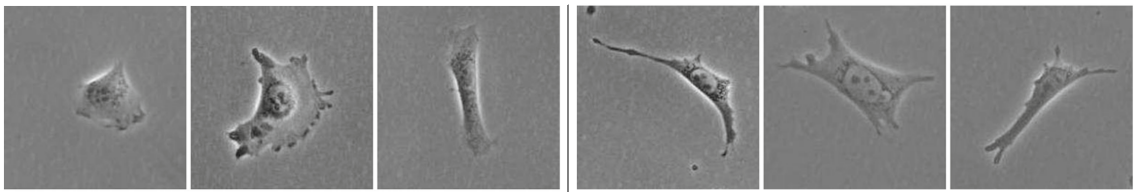


Figure 2.5: *Example of different cell morphologies classified in [150]; cells in the spread and non-polarized (top) and spread and polarized (bottom) states.*

phic metric mapping [64] to characterize the shape distribution of cell nuclei such that would be more flexible than linear methods such as PCA. Singh *et al.* [141] used spherical harmonics-based shape parametrization to represent nuclear morphology[25], encoded with PCA. Wang *et al.* [162] argues that LDA methods, such as Fisher discriminant analysis (FDA) [55], are more appropriate to establish differences between cell population, thus proposed a modification of FDA to characterize differences between nuclei contours.

When the morphology of the cell is simply used to differentiate between cell populations (with a hypothesis known a priori), morphological descriptors have been used in combination with other features to train classifiers. For example, Chaudry et at. [33] used gray-level co-occurrence matrices to encode simultaneously texture and shape and grade renal cancer cells with a Bayesian classifier, and Jones *et al.* [75], from the Cell Profiler team [27], developed an online learning system to classify cell phenotypes using a wide range of texture (e.g. Gabor, entropy, correlation and variance) and shape (e.g. Zernike, eccentricity, form factor and axis lengths) features, with a Gentle-Boosting regression method [59]. More recently, Theriault *et al.* [150] used AdaBoost to classify morphologies in segmented HeLa cells on phase contrast microscopy, using image moments to describe shape, and appearance-based features from the intensity, gradient and Laplacian of the image region.

2.2.2 Characterization of sub-cellular organization

Detailed knowledge of a protein's subcellular location, known as location proteomics, is essential to a complete understanding of its functions [37]. This task can be done in a data-driven fashion by automatically examining the location patterns after protein la-

belling (e.g. using fluorescence in situ hybridization (FISH)) in large data sets. Huang *et al.* [71] explored supervised learning to train discriminative models that could differentiate protein location patterns in 2D and 3D fluorescent images, and compared the performance of eight classifiers, including neural networks, SVMs with different kernels, and ensemble methods such as AdaBoot and Bagging. Different sets of features, termed Subcellular Location Features (SLFs), were extracted from the cell images, which are based on Haralick and Gabor texture features, Daubechies wavelets, Zernike moments, DNA features (relationship between protein fluorescence and DNA fluorescence in parallel images) and geometrical and morphological measurements, and have been widely used in the field. Those features form a feature vector that was then optimized with stepwise discriminant analysis (SDA), achieving a classification accuracy of over 90%. Chen *et al.* [34], added structure to the classification problem solved with inference on a graphical model. In such a graph, each cell is represented by a node with a class probability obtained via an SVM classifier, and edges connecting nodes based on the similarity of their features. This inference aimed to correct for bias in the features for all the cells in each class due to differences in experimental conditions. These supervised methods, however, are applicable when the different patterns are predetermined; these is not always available, thus exploratory methods are required such as clustering algorithms as in [36] and [82]. In order to discover the most discriminative features for this type of exploratory studies, Newberg *et al.* [113] performed a study on a large data set where features such as the common SLF and features relating reference channels (i.e. similarity measures between the object in different image varying the types of labelling) where evaluated. The latter proved to be especially important.

Zhao and Murphy [172] argued that discriminative models cannot capture the location information required in systems biology studies, and proposed a generative model for 2D images, extended to 3D in [120] . In this approach, "objects" within cells (i.e. tagged proteins and organelles) are detected and encoded with texture and shape features, and modelled with multivariate Gaussian distributions as described in [173]. Then, the loca-

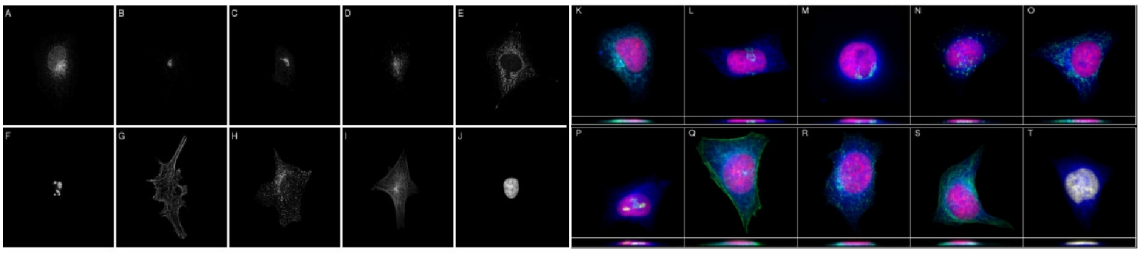


Figure 2.6: *Example of different protein location patterns in HeLa cells under fluorescent microscopy [71]. Ten different patterns are shown in fluorescent images (A-J), and colour-coded (K-T) to show the target protein (green), total DNA (red), and total protein (blue).*

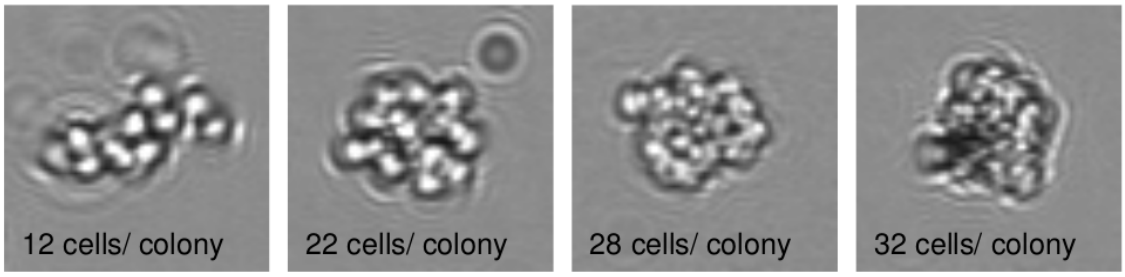


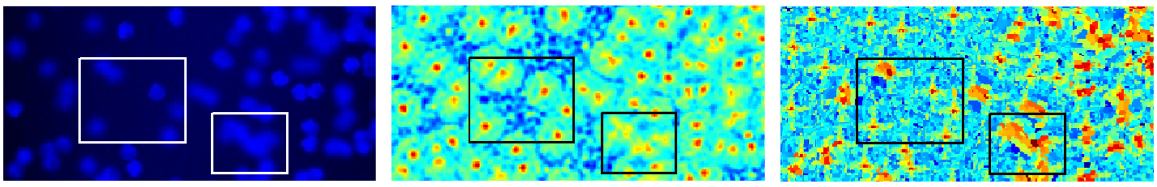
Figure 2.7: *Example of four colonies of approximately same size, but different cell density, taken with a lens-free and large field of view microscopy device used in [56]. Explicit cell detection is not possible for a moderate cell density.*

tion patterns, discovered with clustering algorithms, are represented as Gaussian Mixture models.

The spatial organization of the genome inside cell nuclei has also shown high relevance, since it has a central role in DNA related processes(i.e. repair and replication). Berger *et al.* [19] derived probabilistic maps of spatial arrangements of the genome in eukaryotic cell nuclei by measuring the position of several loci relative to nuclear envelope, the nuclear center and nucleolus; this proved the nonrandom positioning of genes inside cell nuclei.

2.3 Counting

Counting objects in images is generally done directly from detection algorithms, thus all methods described in the detection and segmentation section can potentially suit this objective. However, a few recent approaches have been developed specifically for counting



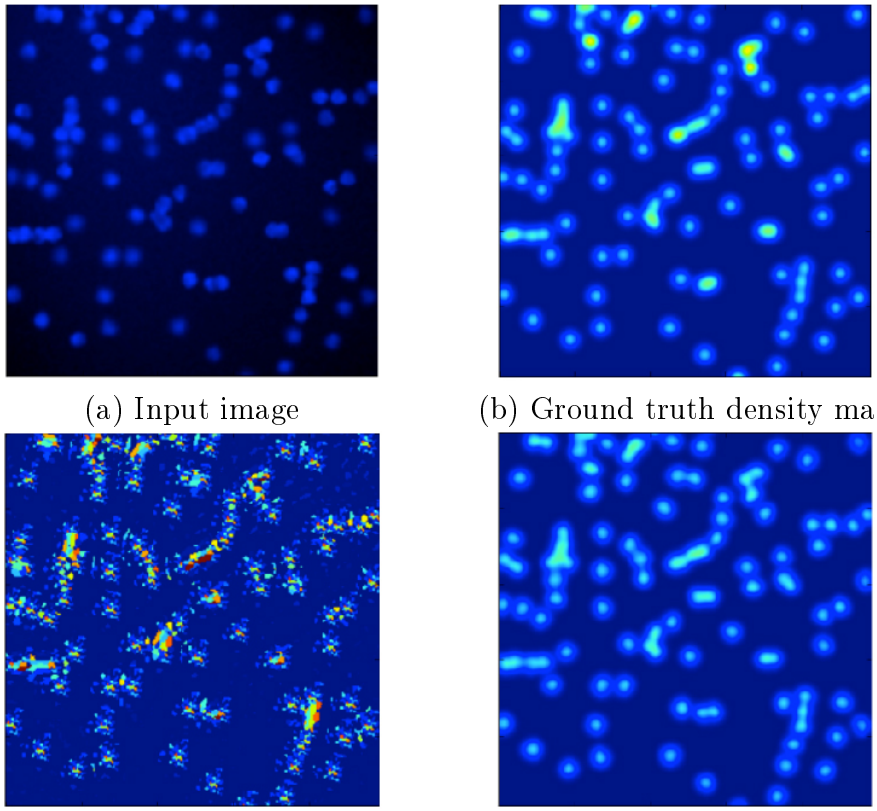
(a) Input: 6 and 10 cells (b) Detection: 6 and unclear (c) Density: 6.52 and 9.37

Figure 2.8: *Example of counting through density estimation from [90]. (a) An input image emphasizes two windows of a synthetic microscopy image containing 6 and 10 cells respectively. (b) The confidence map produced by an SVM-based detector shows 6 peaks clearly discernible for the 1st window, but the number of peaks in the 2nd window is unclear due to the cell overlap. (c) In the density map produced by the density estimation method, the integrals over the rectangles (6.52 and 9.37) are close to the correct number of cells.*

which do not require explicit object detection. This kind of methods target applications where detecting individual instances might not be possible, e.g. when counting cells in very large fields of view (see Figure 2.7), or from images with low resolution such as cases of *in-situ* microscopy or mobile microscopy.

The first attempts and most common methods have been area-based counting [14, 99, 103, 143] and template matching [135, 139]. Although these methods have been useful for certain applications, they depend strongly on heuristics, and are not robust to variable object size and density. Therefore, learning-based methods have emerged with the ability to estimate an object density function whose integral over any arbitrary image region equals the object count in it.

Density estimation methods have been popular in the computer vision community, mainly to count people in crowd analysis, e.g. [29, 72, 80, 90, 97, 128]. The approach consists of collecting image features that are mapped into a object density through a mapping vector learned by regression with neural networks [41, 80, 101], Gaussian process regression [29], linear regression [128], random forest regression [53], or by optimizing a counting-specific cost function [90]. The latter, by Lempitsky and Zisserman, proposes a learning framework that optimizes a loss function based on a distance measure particularly appropriate for a counting task (see Figure 2.8), called MESA distance, and its applicability to cell counting on static images was further demonstrated in [57]. Following [90], Fiaschi *et al.* [53] proposed to learn a density estimator using regression forests



(a) Input image (b) Ground truth density map

(c) Estimated density map of [90] (d) Estimated density map of [53]

Figure 2.9: *Example of density estimation from [53] and comparison with [90]. (a) For an input image of synthetic cells, and (b) its ground truth density map, the figure compares (c) the estimated density map computed with [90] to (d) the one computed with [53]. Even though both methods achieve excellent counting estimates for this dataset, the method of [53] produces smoother estimated density maps, which can facilitate other subsequent tasks such as object localization based on the density estimation.*

which produced smoother density maps, as shown in Figure 2.9.

Density estimation methods are used in Chapter 6 to count objects in both still frames and temporal sequences within different microscopy applications, and an interactive version of the density estimation approach is described in Chapter 7.

2.4 Characterization and visualization of screenings

So far, we have discussed the parsing (e.g. detection, segmentation or counting) of specific individual objects within microscopy images which are relevant throughout Chapters 3-7. We now transition to the more general scenario of assessing whole samples in order to characterize the response of object populations to experimental variables. For example,

consider the case of a large image-based screening which aims to study the effect of a single perturbagen[2] over a specific cell line. In this simple case, the experimentalist might be interested in how a whole cell population changes when the perturbagen is applied relative to when it is not (i.e. a control group), and try to quantify and visualize this difference. Depending on the hypothesized effects of the perturbagen, the image analysis could be focused, for example, in classifying each cell in the samples into a different set of expected phenotypes, followed by a statistical study of the rate of appearance of each of the phenotypes in each of the samples in order to compute an enrichment score (i.e. the likelihood that the concentration of the phenotype of interest is not occurring by chance). Similarly, the analysis could be focused on the population growth dynamics of the cell line (e.g. cell proliferation rate), for which the cell count per sample could be sufficient. Nevertheless, for the case of exploratory studies, a reduced set of hypotheses might not be available, thus more general and potentially highly-dimensional features need to be assessed. For such a case, the image analysis should instead focus on collecting general-purpose visual descriptors in order to discover relevant visual patterns, while allowing the researcher to visualize them in an intuitive way.

This section reviews different techniques used for the characterization of whole populations and the visualization of the patterns in them, that have been used in the high-throughput and high-content screening literature, as well as methods from the computer vision and machine learning communities that can be applied to such a scenario.

2.4.1 Whole sample characterization

In this subsection we review common approaches used to characterize whole samples in high-throughput and high-content screenings. In this context, a sample refers to an entire image or group of images that correspond to the same population; for instance, the images collected from a single well.

The most common approach towards the characterization of a whole sample has been

[2]Perturbagen is a general term used to refer to treatments such as small-molecule compounds or RNA inhibitors, among other.

the combination of individual object data, e.g. individual cells, and it is commonly available in high-throughput screening analysis software. For example, using the open-source CellProfiler Analyst software [76], Jones *et al.* processed a screening that aimed to relate genetic information in Drosophila melanogaster Kc167 cells with perturbations in the cell cycle. A variety of morphological features were collected from individual cells, and then analyzed to identify the different existing phenotypes, which in this case corresponded to different stages of the cell cycle. By calculating the number of nuclei in metaphase- or telophase-stage in each sample, it was possible to identify relevant sub-populations with a perturbed cell cycle from which the genetic information was then studied (i.e. microarray data) in order to identify the potential genetic causes for the differences between populations. Following this approach for sample characterization, several imaging-based studies have proven successful by simply choosing and analyzing different features or readouts according to the phenotypes of interest, such as cell proliferation (i.e. cell count) of human hepatocyte [137] or stem cells [16] as a response to small-molecule libraries or different hypoxia conditions, respectively; fluorescent intensity and morphological and texture features to assess difference in a nucleoplasmic and nucleolar accumulation of a fluorescently tagged protein in a siRNA screen monitoring [69] or the fat accumulation in c. elegans after genetic or chemical perturbations [161]; the shape of c. elegans as a response to small-molecule and bio-active compounds [114]; or even the location and orientation of zebrafish larvae to characterize the response to several genetic and environmental factors [44].

In all of the examples mentioned above, the studies included a hypothesis of the features that needed to be assessed by the image processing algorithms. Moreover, the applicable visual features could be generally interpreted even by non-image analysis experts, facilitating the understanding of the results. Nevertheless, in studies that are mostly exploratory the phenotype hypothesis might not be available, or previously unknown effects on the target can arise during the experimental process, thus creating the need for alternative strategies.

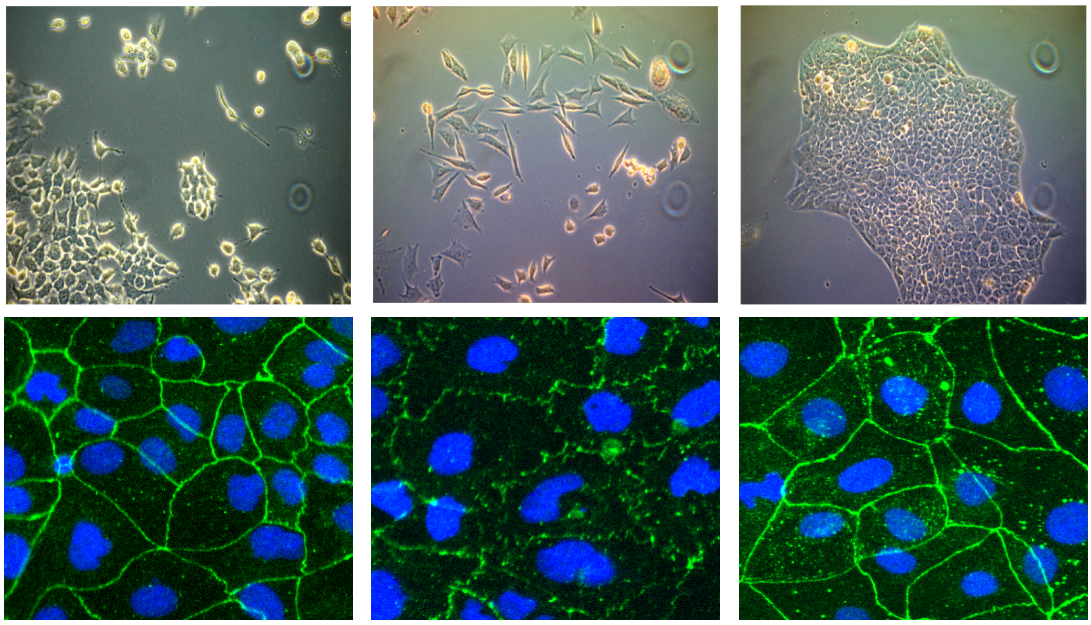


Figure 2.10: *Example of screenings where relevant patterns are not only local but also global, and therefore pure cell-based measurements can fail to capture the entire effect caused by the perturbagens. The top row shows images of epithelial cancer stem cells (CSCs) treated with different agents for epithelial CSC-specific toxicity [65]. The bottom row is taken from a screening on Caco-2 cells used in Chapter 8 from the Ludwig Institute for Cancer Research (University of Oxford, UK), where the ASPP2 protein is tagged in order to explore its reaction to a small-molecule library.*

One appealing strategy is to include an interactive loop between the algorithms and the experimentalist [77, 110], allowing the progressive discovery of the relevant phenotypes; for instance, through active learning. After a finite and manageable space of phenotypes has been identified, the sample characterization can proceed as previously described given the individual object data.

The active learning strategy is particularly valid if the experimentalist is able to objectively identify the relevant phenotypes, but this might not be the case if the necessary visual features are rather abstract, or if the space of phenotypes is too large (see Figure 2.10). For example, the screen could aim to assess the spatial arrangements of the cells within a culture [22,61,85], or as shown in Chapter 8, it can be necessary to capture sample-wise distribution of a tagged protein which appears as complex textures. Capturing such global arrangements objectively and in a reproducible manner, requires the use of algorithms able to handle phenotypes within the high-dimensional space of image-wise

visual features.

Fortunately, the problem of whole image descriptors has been long-researched in the computer vision community, especially within the tasks of image classification and retrieval. The methods for this tasks generally consists of encoding a set of local visual features extracted from the whole image into fixed-length vectors, which make arbitrary images comparable. The common pipeline is illustrated in Figure 2.11. Successful methods within the extensive literature in the area of feature encoding include bag-of-words [142], vectors of locally aggregated descriptors (VLAD) [74] and Fisher encoding [121], which encode the images through different statistics of the local features present in an images with respect to an entire dataset. For instance, in the context of an image-based high-throughput screening, the sample encoding could consist of modelling the screening image data by sampling from the entire dataset in order to build a visual vocabulary (e.g. through k-means) or global distribution of local features (e.g. through a Gaussian Mixture Model), and then representing each sample by collecting its local features and measuring their statistics given the dataset model (e.g. through a histogram of local features or the differences in the distribution of features within the sample with respect to the global distribution). A comparative review of classical encoding techniques for visual features can be found in [30].

Once every sample in the image-based screening is represented by a single vector, it is possible to measure how visually similar two samples are, and even cluster the samples in the screening according to such measure of similarity. In the case of natural images, as commonly used in the computer vision literature, it is often possible to understand what the image encodings capture due to the strong semantic content in the datasets. Therefore, the quality of a dataset clustering could often be qualitatively judged with some objectiveness. In the case of high-throughput screenings, however, the visual patterns within the regimes of interest can be very subtle as well as complex (e.g. consisting of relations between visual patterns). Nevertheless, it is still necessary to interpret the results, formulate hypotheses, and to some extent, provide a mean for quantification,

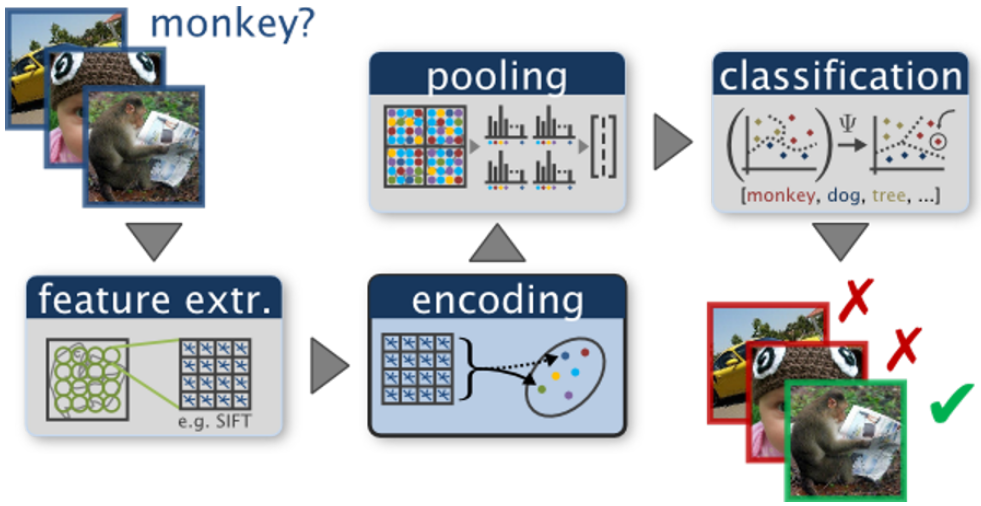


Figure 2.11: *Example of the whole-image encoding pipeline within an image classification task taken from [32]. By doing a global encoding of the local features (e.g. SIFT)it is possible to obtain fixed-length representations of arbitrary input images such that they can be evaluated with a classifier (e.g. an SVM) or retrieve images based on their similarities (e.g. with nearest neighbours)*

thus requiring strong data visualization and grouping methods. We now discuss existing techniques that can be applied for the such task within image-based screenings.

2.4.2 Image data visualization

One aim of the data visualization on a high-throughput screening is that of providing the experimentalist with a full "summary" of the dataset, conveying the idea of the range of effects caused on the target (i.e. the resulting phenotypes) and the relations between the samples (e.g. which perturbagens cause similar effects).

The most intuitive approach towards understanding the relation between the variables measured in a screen is by plotting pairs or triplets of variables together and trying to identify the appearance of different sub-populations, and this is generally implemented in most software packages for high-throughput screening analysis (e.g.[76]). However, this would only succeed if the phenotypes of interest happen to lie in a two- or three-dimensional subspace of the complete set of visual features.

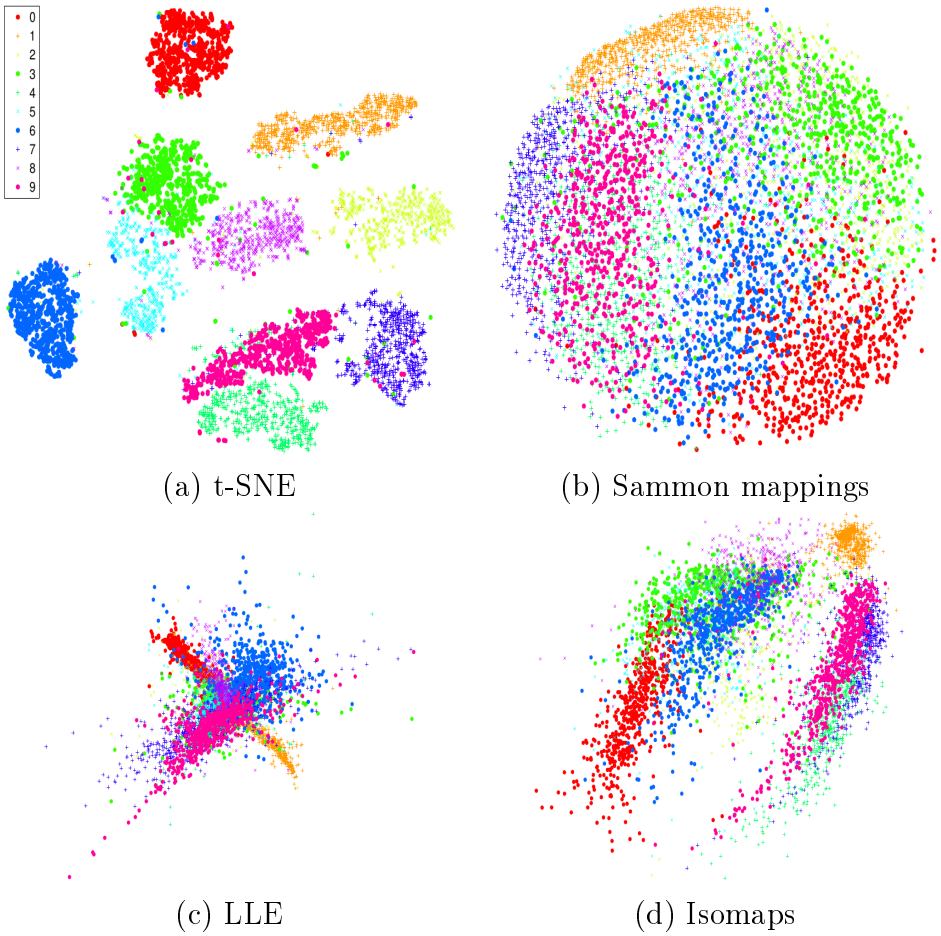
For the cases where the number of features of interest is higher than three, but still of relatively low dimensionality (e.g. lower than 30), it is possible to use more complex and

interesting representation techniques such as glyphs [23], as demonstrated on a breast cancer cell screening in [129]. In this context, the glyphs consist of simplified graphical representations of the cells whose elements vary according to the features extracted from the cell images. Even though comprehensive graphical representations of the different phenotypes can be achieved, a global picture of the similarity between samples still depends on either the human interpretation of the similarity between glyphs, a feature selection procedure to plot over two or three dimensions, or a lower-dimensional projection of the data.

When the descriptors for the samples are highly dimensional, representing them in a way that is easy to interpret becomes even more challenging. Moreover, when the effective dimensionality of the data is higher than the dimensionality of the visualization, there will be a representation error and artifacts that need to be accounted for.

A common way to visualize high-dimensional data is to reduce its dimensionality to two or three dimensions such that it can be represented on a scatter plot. A large variety of methods that have been proposed in the past can achieve this task. Classical dimensionality reduction methods such as multi-dimensional scaling (MDS) [151] and principal component analysis (PCA) [70] can produce the low-dimensional representation of the data. However, linear techniques fail to preserve the structure of the data if this lies in a non-linear manifold. Therefore, non-linear techniques can be more powerful for the visualization task. A few popular examples of non-linear techniques include Sammon mappings [131], isomaps [149], locally linear embeddings (LLE) [127], stochastic neighbour embeddings (SNE) [68], and its extension, t-distributed stochastic neighbour embeddings (t-SNE) [155]. We focus on the latter in our work, as it is a technique especially tailored for visualization and one that tends to outperforms the rest of the example methods in such task, as shown in [155]. An example of a dataset visualization with non-linear dimensionality reduction methods is shown in Figure 2.12.

t-SNE begins by building a distribution P, containing the conditional probabilities $p_{i|j}$ reflecting the pairwise affinities between the high-dimensional vectors that represent the



(a) t-SNE (b) Sammon mappings

(c) LLE (d) Isomaps

Figure 2.12: *Qualitative comparison of different non-linear dimensionality reduction methods for the task of dataset visualization; in this case, the 6,000 handwritten characters from the MNIST dataset [86]. The example is taken from [155]*

samples in the dataset ($A_{i,j}$). An analogous distribution Q is then built for the samples x'_i in the lower-dimensional space, and finally, the values of x'_i are searched such that the Kullback-Leibler divergence between the distributions P and Q is minimized. Due to the quality of the visualizations produced, t-SNE has been widely used in different fields, including bioinformatics (e.g. [4, 7, 13, 17, 92]). We make extensive use of t-SNE in Chapter 8 to facilitate the exploration of image-based high-throughput screenings.

2.5 Summary

This chapter covered a selection of relevant literature on microscopy image analysis to place the contributions of this doctoral research in perspective.

A general trend throughout the different tasks reviewed, and one that is expected, is that learning-based methods tend to offer the flexibility to adapt to different scenarios with relatively little effort from the perspective of the users. Users are often required to simply provide the necessary domain-specific knowledge (e.g. annotations), related to their area of expertise, instead of dealing with possibly more time-consuming parameter tuning or even coding. Therefore, the efforts within the learning-based methods are in how to make algorithms more reliable (e.g. robust and accurate) while minimizing the amount of time or effort required from the potential end users in the non-image analysis communities (e.g. weaker and/or less annotations, and with tolerance to errors).

Chapter 3: Detection of multiple instances of an object in microscopy images

In this chapter we introduce a learning-based method for detection of objects in microscopy images that is general enough to perform well across a variety scenarios such as different microscopy modalities (e.g. fluorescence and phase contrast), or different objects in microscopy images (e.g. cancer cell lines or molecular colonies).

The learning procedure is designed to learn to detect objects from dot-annotated images; that is, where a dot is placed inside each object of interest in the training images (e.g. inside each cell). Given only this minimalistic annotation, the method is able to learn a model such that the objects of interest can then be detected on unseen images provided that a similar experimental setup is maintained. The method is evaluated on several dot-annotated microscopy datasets and it achieves excellent detection accuracy in different scenarios, despite considerable variation between the datasets.

The method uses a highly-efficient Maximally Stable Extremal Region (MSER) detector [104] to retrieve a large number of candidate regions, each of which is assessed with a learning-based measure to determine how similar they are to the objects of interest. A non-overlapping subset of those regions with high similarity to the class of interest can then be selected via dynamic programming, while the model learning can be done within a structured output framework [152].

Before describing the detection method, we introduce the datasets and metrics used to evaluate its performance in Section 3.1, which are also used in Chapter 4 and Chapter 5 for the evaluation of the extensions of the base detection method presented in this chapter.

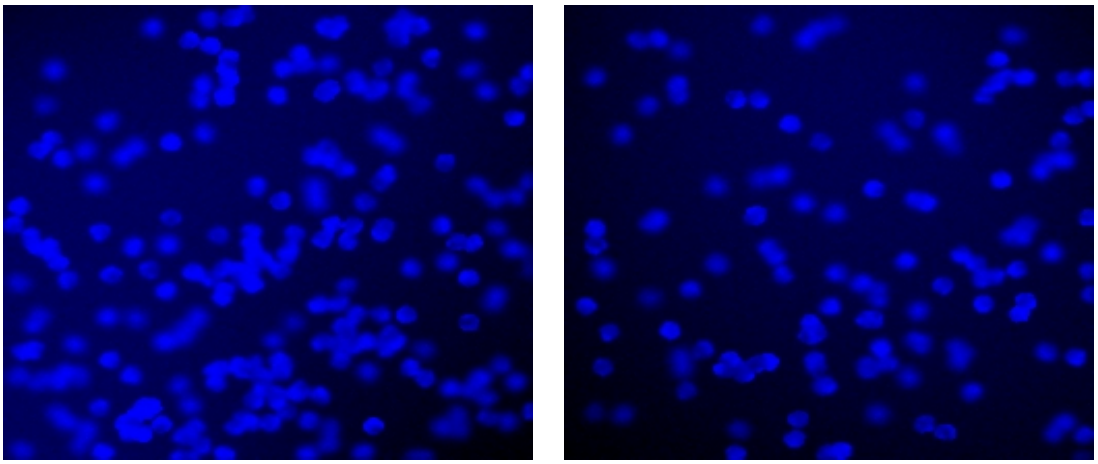
The collection of candidate regions for object detection is then described in Section 3.2, the inference procedure to select the best subset of candidates is described in Section 3.3, and the learning process to assess the quality of candidate regions to guide the inference is described in Section 3.4. The experiments on object detection are presented in Section 3.5, with a conclusion in Section 3.6.

3.1 Datasets

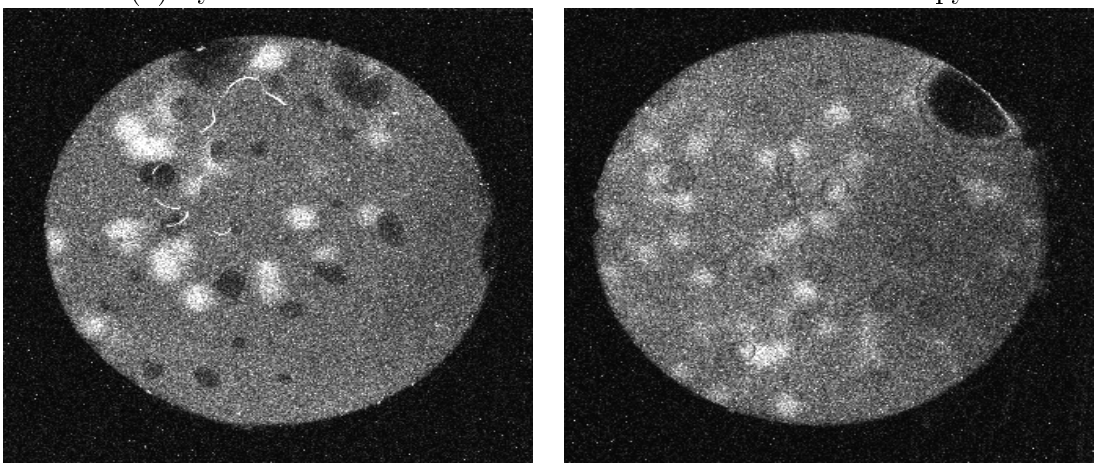
The datasets consist of six distinct microscopy datasets spanning different modalities and imaging conditions. For each of the datasets, the data used for training is divided into several random splits, which are later used to compute means and standard deviations of the evaluation metrics. The task, in all cases, is to detect all instance of the objects (e.g. cells) which have been annotated with dots on a training set. Similarly, the testing sets have been dot-annotated for the purpose of performance evaluation.

The metrics used for the evaluation of the different concepts and methods throughout the detection work are the following: the mean counting error (MCE), which measures instance counting accuracy for a dataset with N images as $MCE = \frac{1}{N}\sum_{i=1}^{N} \mid \hat{c}_i - c_i \mid$, and the $F_1 score = 2 * \frac{precision * recall}{precision + recall}$, which measures instance detection accuracy. Precision and recall are defined in terms of true positive (TP), false positive (FP) and false negative (FN) detections in the following way: $Precision = TP/(TP + FP)$ and $Recall = TP/(TP + FN)$. The assessment of the predictions within a testing image is done by matching the predicted object centroids with the ground-truth dot-annotations using the Hungarian algorithm subject to the constraint that a predicted centroid must lie within a radius ρ of a ground-truth dot. For each dataset, ρ is set to be the average radius of the single objects. Matched pairs of predicted centroids and ground-truth dots are considered true positives, unmatched predictions are considered false positives, and unmatched dot-annotations are considered false negatives.

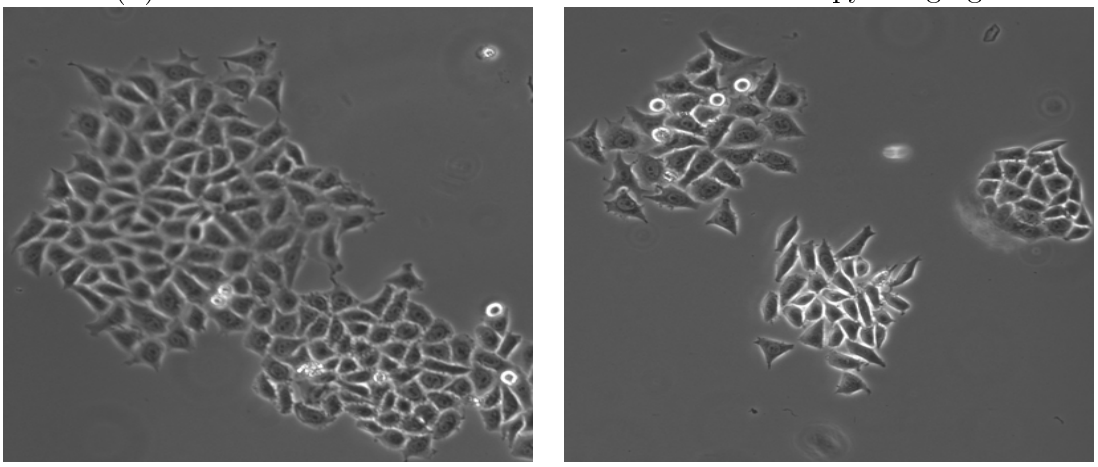
Synthetic fluorescence microscopy (Figure 3.1a). The synthetic dataset [90] represents a good benchmark for comparison of cell detection and counting methods as it



(a) Synthetic dataset of cell nuclei in fluorescence microscopy

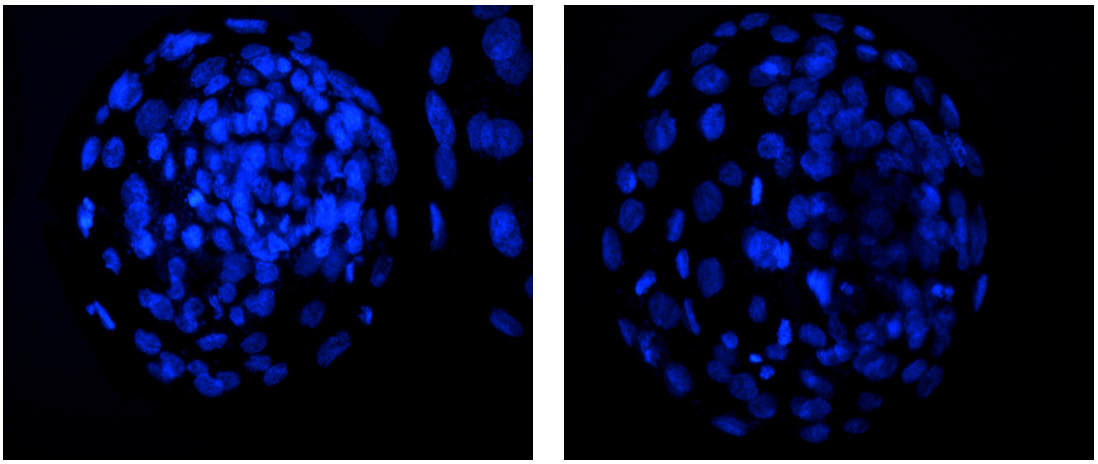


(b) Molecular colonies in weak-fluorescence microscopy imaging

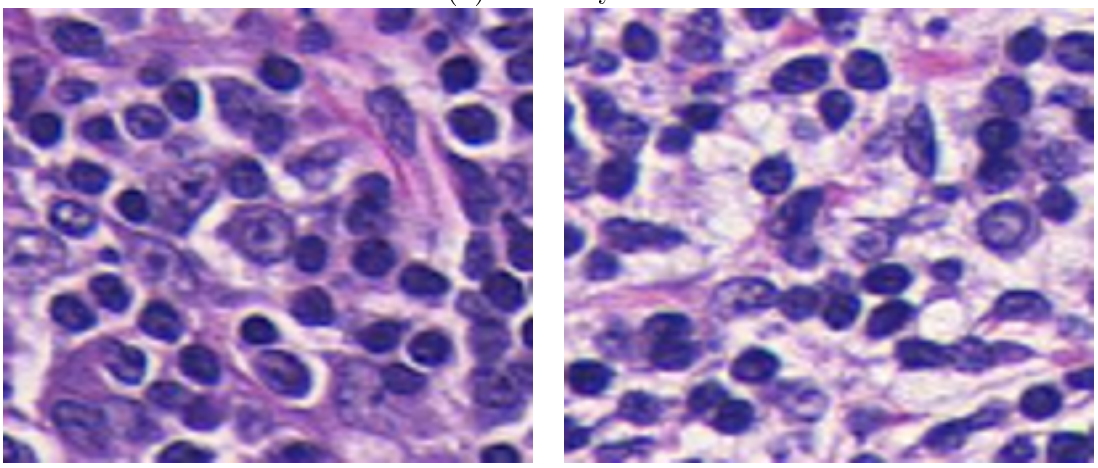


(c) HeLa cell-line in phase contrast microscopy

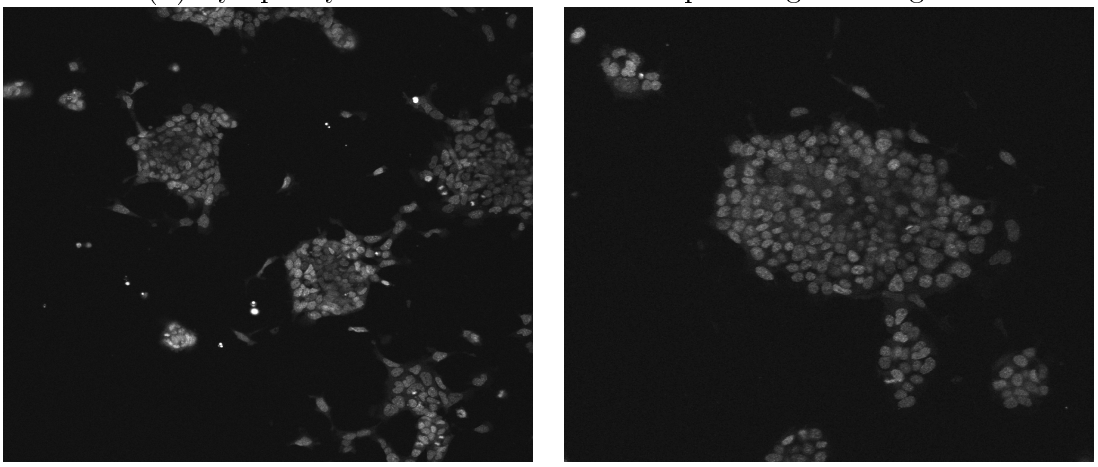
Figure 3.1: *Example images from some of the microscopy datasets used throughout the evaluation of the detection methods. Examples of the remaining datasets can be seen in Figure 3.2. See Section 3.1 for details.*



(a) Blastocysts



(b) Lymphocytes in breast cancer histopathological images



(c) Cell nuclei in fluorescence microscopy

Figure 3.2: *Example images from some of the microscopy datasets used throughout the evaluation of the detection methods. Examples of the remaining datasets can be seen in Figure 3.1. See Section 3.1 for details.*

contains perfect ground truth annotations due to its synthetic nature. It consists of 200 images of cell nuclei on fluorescence microscopy generated with [88]. This dataset can contain severe overlap between instances, which makes it challenging for detection-based methods and more appropriate for counting-based methods. The synthetic dataset is divided into 100 images for training and 100 for testing, and several random splits of the training set are proposed in [90]. Such splits consist of five sets of N training images and N validation images, for $N = 1, 2, 4, 8, 16, 32$.

Weak-fluorescence molecular imaging (Figure 3.1b). The molecular dataset, kindly provided by researchers of the Laboratory for Viral RNA Biochemistry, Institute of Protein Research RAS, consists of images of gels with DNA colonies obtained through *in vitro* amplification [40] (the method is also known as a "polony" technique [107]). Each colony represents a progeny of a single molecule that contains a certain nucleotide sequence. The images were obtained using a confocal microchip laser scanner (PerkinElmer ScanArray Express). Automated counting in this case could enable fully-automatic and real-time monitoring of molecular colonies [130]. While in some circumstances (e.g. diagnostics based on marker RNA) high counting accuracy might not be needed, in other cases (e.g. measuring gene expression) achieving high counting accuracy ($<$20%) is of great importance. This dataset consists of 198 images with shot-noise and low contrast characteristic of weak fluorescence, which poses an additional challenge for methods based on blob detection. As in the synthetic dataset, the molecular data is divided in half for training and testing. We further split the training set into five different random groups consisting of 60 training images and 30 validation images each.

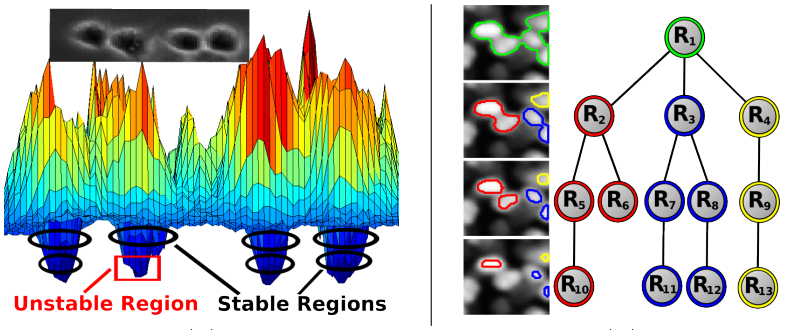
HeLa cells on phase contrast microscopy (Figure 3.1c). This dataset introduced in [9] consists of 22 phase contrast images of HeLa cell cultures, and it is a subset of a control set collected for detailed colony growth monitoring in radiation experiments, with the kind support of Dr. Boris Vojnovic and James Thompson (Grey Institute for Radiation Oncology and Biology, University of Oxford, UK). The HeLa dataset is split into 11 images for training and 11 for testing. Due to the limited amount of training

data, training and validation is done on a leave-one-out fashion following [9].

Blastocysts (Figure 3.2a). Cell number in *in vitro* produced blastocysts is one of the important parameters for estimation of embryo developmental potential, and thus, oocyte quality. The cell count at different times of *in vitro* embryo development is routinely used in the research targeting the improvement of assisted reproduction technologies both for animals and for humans. Labeling of cell nuclei by fluorescent dyes binding to a double-stranded DNA is routinely used method to visualize either fixed or living cells. This dataset, kindly provided by Dr. Svetlana Uzbekova (INRA, Physiology of Reproduction and Behavior Unit, Nouzilly, France), contains 22 images of the outer cell layer of blastocysts. The images in this dataset show severe cell overlap resulting from the projection of the blastocysts (spheres) into a 2D image, making the individual cell detection task quite challenging. Still, 2D microscopy is a popular tool for this task due to its much lower cost compared to 3D microscopy, and the tendency of the majority of the cells (so called *inner cell mass*) to concentrate on one side from the blastocyst cavity thus allowing analysis after the projection to 2D. We divide the training data into 5 random splits, consisting of 8 training images and 3 for validation.

Lymphocytes in histopathology (Figure 3.2b). The histopathology dataset was introduced in [66] and the task is the detection of lymphocytes on stained breast cancer tissue, which is a prognosis indicator for various types of breast cancers. The main challenge of the task comes from the fact the lymphocytes need to be discriminated from the cancer cells, which have very similar appearance. The dataset consists of 20 images and is divided in half for testing and training, and five random splits of 8 and 2 images for training and validation are used in the experiments.

Cell nuclei on fluorescence microscopy (Figure 3.2c). The final dataset, kindly provided by Dr. Julian Gingold, is another real example of fluorescence microscopy where cell nuclei need to be detected. The images correspond to RNA interference experiments on mouse embryonic stem cells, where single cell detection is required for further processing in order to characterize cell changes in the population as a response to different



(a) (b)

Figure 3.3: *(a) Example of the intensity profile of an image region containing cells. The MSER algorithm detects extremal regions that are stable in area growth while varying an intensity threshold. Typically, many extremal regions are nested within and between cells (especially when there is cell clumping) forming a tree structure. For example, (b) the boundaries of several MSERs that appear in the close-up of a cell image are shown, which can be represented by the tree structure. The parent-child relationships in the tree correspond to the nestedness of the regions. The tree structure is utilized by the inference algorithm.*

experimental conditions. Partial cell overlap and cells slightly out of focus pose the main difficulties for nuclei detection in these images. The dataset consists of 20 images and, once more, is divided in half for training and testing, where five random splits of 8 and 2 images for training and validation are used in the experiments.

3.2 Candidate regions for detection

The detection model operates by first producing a set of overlapping candidate regions, and then picking a subset of those regions based on a learned classifier score and subject to a *non-overlap constraint*. We consider the use of *extremal regions* as candidates for cell detection.

Extremal regions of the grey-value image $\mathcal{I}$ are defined as connected components of a thresholded image $\mathcal{I}_{>t} = \{\mathcal{I} > t\}$ for some t. In other words, a region is extremal if the image intensity everywhere inside of it is higher than the image intensity at its boundary. Our approach thus builds on the fact that in many microscopy modalities, cells show up as bright or dark blobs in one of the intensity channels, and therefore can be closely approximated by extremal regions of this intensity channel. An important property of extremal regions is their *nestedness*, i.e. the fact that for the same image $\mathcal{I}$ two extremal

regions R and S can be either nested or non-overlapping ($R \subset S$ or $R \supset S$ or $R \cap S = \emptyset$. Therefore, when representing a set of extremal regions as a graphical model, the result is a tree-structured graph that is convinient for efficient inference (see Figure 3.3).

The number of extremal regions on an image can be very large, so in practice we consider only regions that are *maximally stable* in the sense of [104], i.e. the speed of their area variation w.r.t. changing threshold t is a local minimum and is below a separate *stability threshold*. We thus use a popular and efficient maximally stable extremal region detector (MSER) [104] to find a representative subset of all extremal regions. To boost the recall for cell detection, we set the stability threshold to a very high value, so that the MSER-detector produces a manageable but very large (thousands) number of candidate regions. Our inference procedure then determines which of those candidates correspond to objects of interest.

Using extremal regions as candidates for detecion of objects in microscopy images can also have drawbacks. A first major failure mode comes from object overlap. If two objects are overlapping, it will be likely that there will not exist extremal regions that represent each of the objects individually, thus complicating the instance detection task. A second mode where the extremal regions can result in poor detection candidates is when image noise (i.e. due to poor contrast) breaks the assumption that the intensities inside the objects of interest are always higher or lower than outside of it. Nevertheless, as shown in this chapter, there are common scenarios in microscopy imaging where these failure modes do not occur, and extremal regions can be directly used as detection candidates with excellent results. Furthermore, the failure modes due to the main deficiencies of the candidate regions are addressed in subsequent chapters.

3.3 Inference under a non-overlap constraint

Let $R_1, R_2, \ldots R_N$ be the candidate set of N extremal regions detected in an image. Let us assume that each region R_i is assigned a value V_i, which is produced by a classifier and indicates the appropriateness score of this region to the class of cells we want to

detect. Our method then picks a subset of extremal regions so that the sum of scores of the picked regions is maximized, while the picked regions do not overlap (the non-overlap constraint). To formalize this task, we define a set of binary indicator variables $\mathbf{y} = \{y_1, y_2, \dots y_N\}$ so that $y_i = 1$ implies that the region R_i is being picked. Let $\mathcal{Y}$ be a set of those region subsets that do not have region overlap, that is, $\mathcal{Y} = \{\mathbf{y} \,|\, \forall\, i, j : \; (i \neq j) \wedge (y_i = 1) \wedge (y_j = 1) \Rightarrow R_i \cap R_j = \emptyset\}$. Then, the optimization task faced by the model is:

$$F(\mathbf{y}) = \max_{\mathbf{y} \in \mathcal{Y}} \sum_{i=1}^{N} y_i \, V_i \;\; . \tag{3.1}$$

For an arbitrary set of regions, maximizing (3.1) over $\mathbf{y} \in \mathcal{Y}$ is NP-hard (equivalent to *submodular maximization*). However, the maximization of (3.1) can be performed exactly and efficiently by exploiting the nestedness property of the region pool. Indeed, one can consider the tree-structured model, where each node corresponds to a region and where parent-child links correspond to the nestedness relation (Figure 3.3b). Namely, the node R_j becomes a parent of the node R_i if R_j is the smallest region in the pool that R_i strictly belongs to. In this way, the region pool can be organized into a set of trees (i.e. a forest). The idea is then to realize the scores and the non-overlap constraints using pairwise terms of a graphical model with the topology defined by the region trees.

This is achieved using the same trick as in Lempitsky *et. al* [89]. We introduce the auxiliary variables z that are uniquely determined by the initial variables $\mathbf{y}$ in the following sense: $z_i = 1$ iff either $y_i = 1$ or some y_k such that R_k is an ancestor of R_i in the tree equals 1 (note that two ancestors of the same region cannot be assigned non-zero labels simultaneously as long as $\mathbf{y} \in \mathcal{Y}$). The optimization (4.1) can then be rewritten as a pairwise tree-structured MRF on the auxiliary variables:

$$F(\mathbf{z}) = \max_{\mathbf{z}} \sum_{i | p(i) \neq 0} W_i(z_i, z_{p(i)}) + \sum_{i | p(i) = 0} V_i(z_i), \tag{3.2}$$

where $p(i)$ maps region R_i to the number of its parent region ($p(i) = 0$ for root regions in the forest), $W_i(1,1) = 0$, $W_i(1,0) = V_i$ and $W_i(0,1) = -\infty$.

After such variable change, all $y \in Y$ are one-to-one mapped to z configurations with the finite values of the functional (3.2) and this mapping preserves F. Indeed, the infinite terms in W_i enforce the monotonicity of the labelling (from the root to the leaves), meaning that along each path from the root to a leaf the first t (potentially t=0) nodes are assigned $z_i = 0$ and the rest (potentially zero) nodes are assigned the constant label 1. This corresponds to the labeling that assigns y_i=1 to the (t+1)'th node along the path and y_i=0 to all other nodes along the path. As a result, no overlaps are possible between the regions with $y_i \neq 0$ (since each path in the tree has at most one node with $y_i \neq 0$). The non-infinite terms $W_i(z_i, z_{p(i)})$ at the non-root nodes encode the terms $V_i(y_i)$ in the original functional (3.1) (once again, the monotonicity of the labeling z along any path from the root to a leaf ensures that at most one non-zero non- infinite term $W_i(z_i, z_{p(i)})$ corresponding to $y_i \neq 0$ is present within such path).

The optimization task (3.2) can be accomplished via tree-based dynamic programming [118] (the max-sum version of the algorithm). It is then trivial to compute the optimal solution of (3.1) from the optimal solution of (3.2).

3.4 Learning formulation

As discussed in Section 3.3, our method relies on machine learning to score each region for the detection task. A suitable scoring can be learned in a principled fashion from the dot-annotated training data as follows. Assume a set of M training images $\mathcal{I}^1, \mathcal{I}^2, \ldots \mathcal{I}^M$, where each training image $\mathcal{I}^j$ has a set of N^j MSER regions $R_1^j, R_2^j, \ldots R_{N^j}^j$. For each of these regions R_i^j a feature vector $\mathbf{f}_i^j$ is computed (the feature vector choice is described in the implementation details). Finally, assume that the images are annotated, so that n_i^j denotes the number of user-placed dots (annotations) inside the region R_i^j. To obtain the score for each region, we use a linear classifier so that the value V_i^j for the region R_i^j is computed as a scalar product $(\mathbf{w} \cdot \mathbf{f}_i^j)$ with the *weight vector* $\mathbf{w}$. The goal of learning is then to find a weight vector so that the inference procedure tends to pick regions with $n_i^j = 1$, and also to ensure that, for each dot, a region is picked that contains it. In this

way, the produced set of regions tends to be in a one-to-one correspondence with the user-placed dots.

Learning via binary classification. The simplest way to learn $\mathbf{w}$, and one that already produces competitive results in our comparisons (Figure 3.5), is to learn a binary classifier. For this, all regions in the training images are considered, and those with $n_i^j = 1$ are assigned to the positive class while all others are assigned to the negative class. Training any linear classifier, e.g. via a support vector machine algorithm [43], then produces a desired $\mathbf{w}$.

Structured learning. Learning via binary classification does not take into account the non-overlap constraint. A more principled approach is to use a structured SVM [152] that directly optimizes the performance of the inference procedure on the training set. Consider the configuration $\mathbf{y}^j \in \mathcal{Y}^j$ defining the set of non-overlapping regions for the image $\mathcal{I}^j$. It is natural to define an error measure (the *loss*) associated with $\mathbf{y}^j$ as the deviation from the one-to-one correspondence between the user-placed dots and the picked regions:

$$L(\mathbf{y}^j) = \sum_{i=1}^{N^j} y_i^j \, |n_i^j - 1| + U^j(\mathbf{y}^j) \tag{3.3}$$

$U^j(\mathbf{y}^j)$ denotes the number of user-placed dots that are not covered by any region R_i^j with $y_i^j = 1$ (i.e. have no correspondence).

To perform the learning, the "ground truth" configuration $\bar{\mathbf{y}}^j = \{\bar{y}_1^j, \bar{y}_2^j, \ldots \bar{y}_{N^j}^j\} \in \mathcal{Y}$ is defined for each training image by assigning a unique extremal region to each dot (see implementation details). The structured SVM method [152] then finds the optimal weight vector $\mathbf{w}$ by minimizing the following convex objective:

$$\mathcal{L}(\mathbf{w}) = \frac{1}{2}||\mathbf{w}||^2 + \frac{C}{M} \sum_{j=1}^{M} \max_{\mathbf{y}^j \in \mathcal{Y}^j} \left(\sum_{i=1}^{N^j} (\mathbf{w} \cdot \mathbf{f}_i^j) \; , y_i^j - \sum_{i=1}^{N^j} (\mathbf{w} \cdot \mathbf{f}_i^j) \, \bar{y}_i^j + L(\mathbf{y}^j) \right) \tag{3.4}$$

where the first term is a regularization on $\mathbf{w}$, C is a scalar regularization parameter, and the maximum inside the sum represents a convex (in $\mathbf{w}$) upper bound on the loss (3.3),

that the inference (3.1) incurs on the jth training image [152].

The objective (3.4) can be optimized with a standard cutting-plane algorithm [152] provided that it is possible to perform the *loss-augmented inference*, which corresponds to finding maxima inside the second term of (3.4) for a fixed $\mathbf{w}$. Thus, one needs to solve:

$$\max_{\mathbf{y}^j \in \mathcal{Y}^j} \left(\sum_{i=1}^{N^j} (\mathbf{w} \cdot \mathbf{f}_i^j)\, y_i^j - \sum_{i=1}^{N^j} (\mathbf{w} \cdot \mathbf{f}_i^j)\, \bar{y}_i^j \;;\; + \;\; \sum_{i=1}^{N^j} y_i^j \, |n_i^j - 1| + U^j(\mathbf{y}^j) \right) \tag{3.5}$$

We then note that under the non-overlap constraint, the number of un-matched dots $U^j(\mathbf{y}^j)$ can be rewritten as $D^j - \sum_{i=1}^{N^j} y_i^j n_i^j$, where D^j is the total number of dots in the jth training image. After substituting $U(\mathbf{y}^j)$ and omitting the terms independent of $\mathbf{y}^j$, an equivalent optimization problem is obtained:

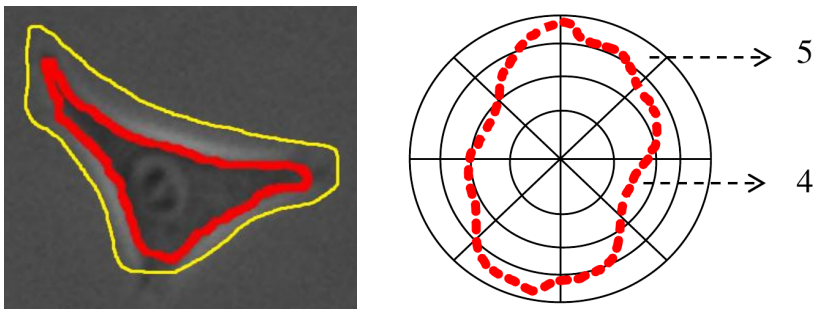
$$\max_{\mathbf{y}^j \in \mathcal{Y}^j} \sum_{i=1}^{N^j} \left((\mathbf{w} \cdot \mathbf{f}_i^j) + |n_i^j - 1| - n_i^j \right) \, y_i^j \tag{3.6}$$

which has exactly the same form as (3.1) with $V_i = (\mathbf{w} \cdot \mathbf{f}_i^j) + |n_i^j - 1| - n_i^j = (\mathbf{w} \cdot \mathbf{f}_i^j) - [n_i^j \geq 0]$. Thus, we can perform loss-augmented inference exactly via dynamic programming on trees, and get an optimal $\mathbf{w}$ through the cutting-plane procedure [152].

To generate the ground truth configuration for the structured learning, we first score all regions using the weight vector w_{bin} learned through a binary SVM. Then, for each dot, we include into the ground truth configuration the region that contains only this dot and has the highest score.

3.5 Experiments

In all experiments, the feature vector $\mathbf{f}_i^j$ used to encode each candidate region consists of the concatenation of descriptors that aim to characterize the size, shape, colour (or intensities) and information regarding the local context of regions. The specific dimensions of the descriptors were chosen empirically, but the method is not sensitive to such choices. In detail, a feature vector concatenates the following descriptors: (i) a 150-dimensional



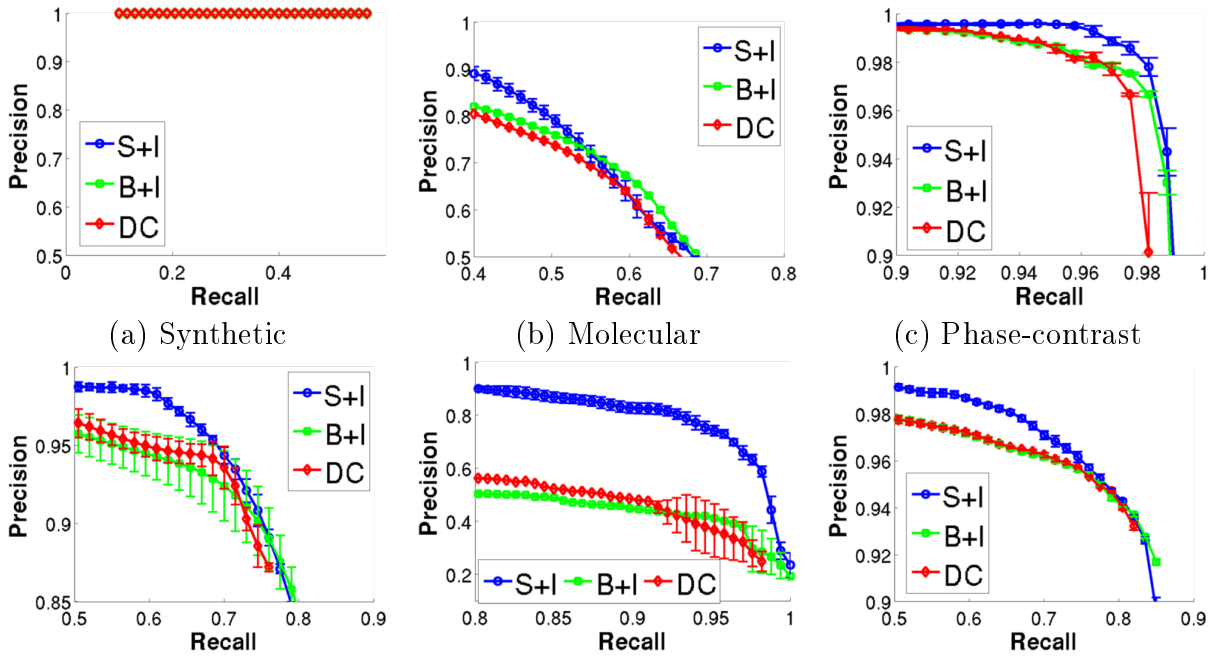
(a) Context (b) Shape

Figure 3.4: *Illustration of two region descriptors. (a) The context descriptor aims to capture information about the surroundings of a candidate region, for which it computes the histogram of differences between the intensities of the pixels that lie in the boundary of the region (red curve) and the pixels that lie in a dilation of such boundary (yellow curve). In this example, if the candidate region matches the cell boundaries, the differences w.r.t. to its dilation will be large. (b) The shape descriptor captures the shape of the boundary of the candidate region through a histogram. The histogram is built by placing the candidate region, which size-normalized and aligned to its major axis, within a polar coordinate system binned in angles and radii. Then, the count of every bin corresponds to the pixels of the boundary that fall within it.*

vector soft-encoding the size of the region by setting to 1 the entry corresponding to its size (in pixels), and then smoothing the resulting binary vector with a Gaussian kernel in order to allow for size variability, (ii) a 12-dimensional histogram of pixel intensities inside the candidate region, (iii) two 8-dimensional histograms of difference of intensities between the boundary of the extremal region and a dilation of it (over two different dilation scales) in order to capture contextual information of the candidate regions (i.e. how it compares to its surroundings; see Figure 3.4a), and finally, (iv) a shape descriptor, similar to *shape context* [18], represented by a 60-dimensional histogram of the distribution of the boundary of the region on a size-normalized polar coordinate system (see Figure 3.4b).

As an indication of the running time, detecting cells on a 400-by-400 pixel phase contrast image takes 30 seconds on a single core of an i7 CPU (dominated by our unoptimized MATLAB code for feature computation).

Three variations of the detection method are evaluated on all the microscopy datasets: (I) *direct classification (DC)*, which evaluates all candidate regions with a $\mathbf{w}$ vector learned via a binary classifier and chooses the region with the highest score in every set of overlap-



(a) Synthetic (b) Molecular (c) Phase-contrast

(d) Blastocysts (e) Histopathology (f) Nuclei in fluorescence

Figure 3.5: *Precision (vertical) vs Recall (horizontal) curves for the six microscopy datasets of Section 3.1 and the three variations of our approach. Significant improvements brought by the non-overlap constraint under the structured SVM (S+I) can be observed.*

ping regions with positive scores, (II) *binary SVM + inference (B+I)*, which does the full inference (3.1) based on the weight vector learned through binary classification, and (III) *structured SVM + inference (S+I)*, which uses inference with the weight vector learned by the structured SVM (3.4).

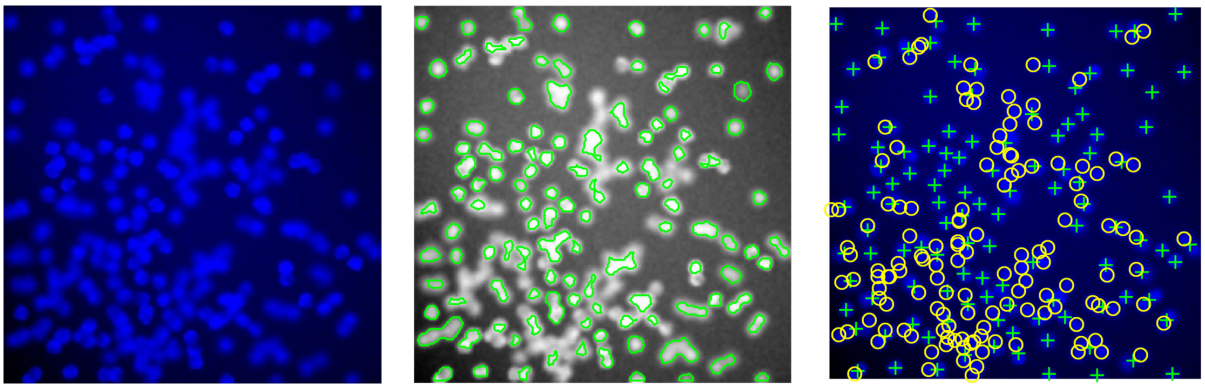
Example results of the (S+I) variant are shown in Figures 3.6, 3.7, 3.8, 3.9, 3.10 and 3.11. Qualitatively, it is possible to see that for the simple cases (i.e well-defined single cells), the detection method is able to detect the instances of the objects, and also to select regions that very well segment the objects of interest. It is also noticeable that clusters are generally discarded (e.g. cells in Figure 3.6) as the candidate regions in them do not correspond to single instances. Finally, in Figure 3.7 it can be seen that the candidate regions are highly irregular, which not only match the objects poorly, but also increases the difficulty of the learning.

Figure 3.5 shows the precision-recall curves for the three variations of our detection method, computed in the following way: a constant τ was added to the score of each

region, and several precision and recall operating points were obtained by varying τ; the precision values at a fixed set of recall points were obtained through spline interpolation, and finally, the mean (plotted point) and standard deviation (error bar) were computed for each recall point using the precision value of each of the different splits of the training data in each dataset.

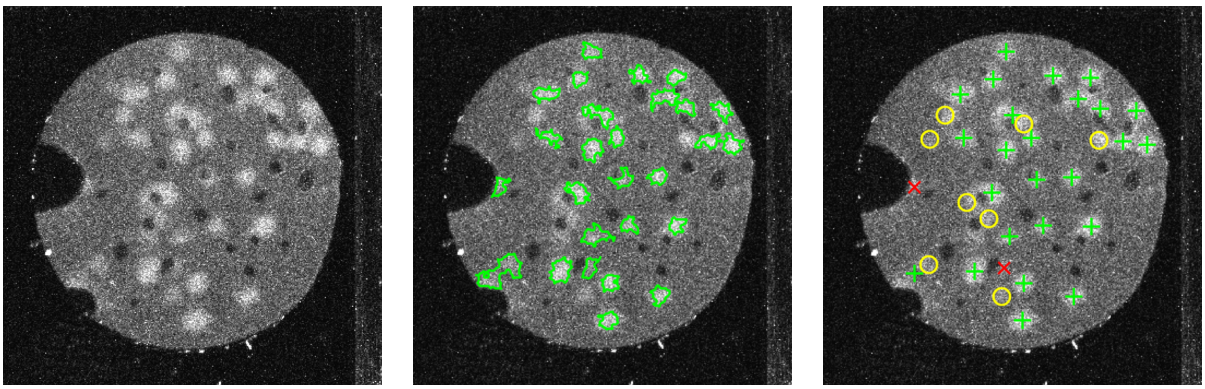
In general, it can be seen that enforcing the non-overlap constraint can increase the accuracy of the method, but it does it considerably better when $\mathbf{w}$ is learned within the structured SVM framework. However, it is clear that in the histopathology, phase contrast and fluorescence datasets, the performance is better than in the remaining three datasets: synthetic, blastocysts and molecular imaging. The underlying reason for this difference is the amount of overlap between the instances of the objects of interest. The way the instance overlap affects the detection method rests on the candidate regions; the learning assesses if each region corresponds to a single object of interest or not, and thus, it requires at least a candidate region per object instance. This assumption might not occur when the instances overlap, and it gets worse as the instance-overlap increases. An extreme case can be seen in the synthetic dataset, where a cluster of objects can contain as many as 7 or more instances within a single candidate region. Nevertheless, for the applications where instance overlap is not frequent, the detection method for individual instances of the objects (singletons) can produce excellent results. Therefore, we now focus on the comparison and further analysis of the detection method against methods in the literature only for the datasets that fall within this category: histopathology and phase contrast. The remaining datasets are further evaluated in subsequent chapters (Chapter 4 and Chapter 5), where the detection method is extended.

Comparison with state of the art. Table 3.1 compares our experimental results (S+I variant) on the histopathology dataset to the methods presented in the ICPR 2010 contest, and to [84] and [20], published since then. The overall comparison is favourable to our method, with a considerable improvement on precision and recall over all other methods. Nevertheless, we note that that the performance in the such dataset is affected by



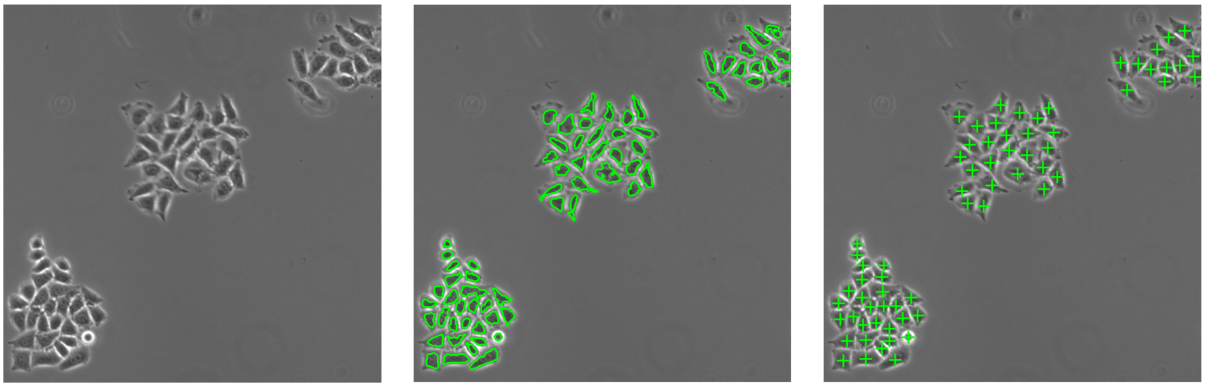
Original Selected regions Detections

Figure 3.6: *Example results of the (S+I) variant on the synthetic dataset. The selected regions (*middle*) in this image should correspond to single instances of the synthetic cells. In the detection image (*right*), correct detections are denoted with a green '+', false detections with a red 'x' and missed instances with a yellow 'o'.*



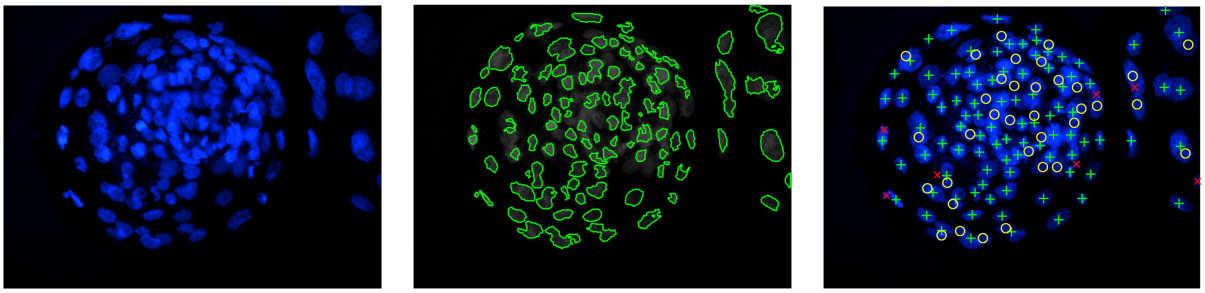
Original Selected regions Detections

Figure 3.7: *Example results of the (S+I) variant on the molecular dataset. The selected regions (*middle*) in this image should correspond to single instances of the molecular colonies. In the detection image (*right*), correct detections are denoted with a green '+', false detections with a red 'x' and missed instances with a yellow 'o'.*



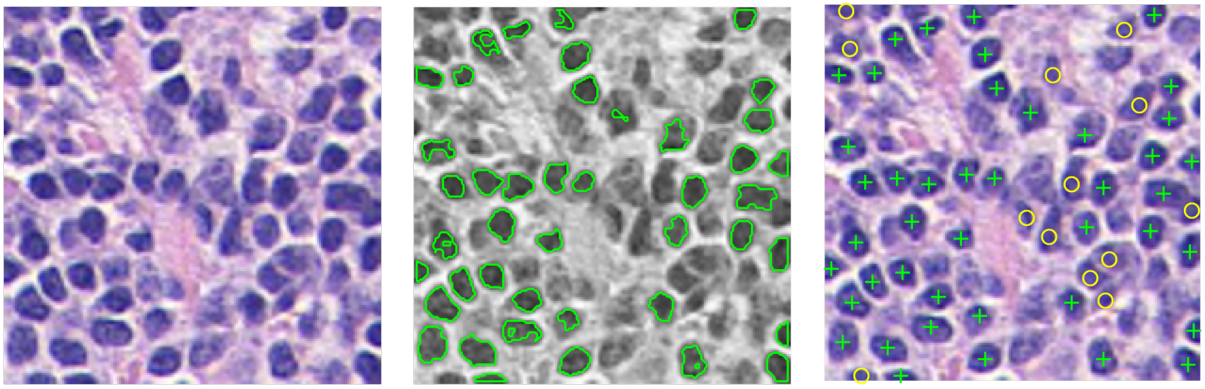
Original Selected regions Detections

Figure 3.8: *Example results of the (S+I) variant on the phase contrast dataset. The selected regions (*middle*) in this image should correspond to single HeLa cells. In the detection image (*right*), correct detections are denoted with a green '+', false detections with a red 'x' and missed instances with a yellow 'o'.*



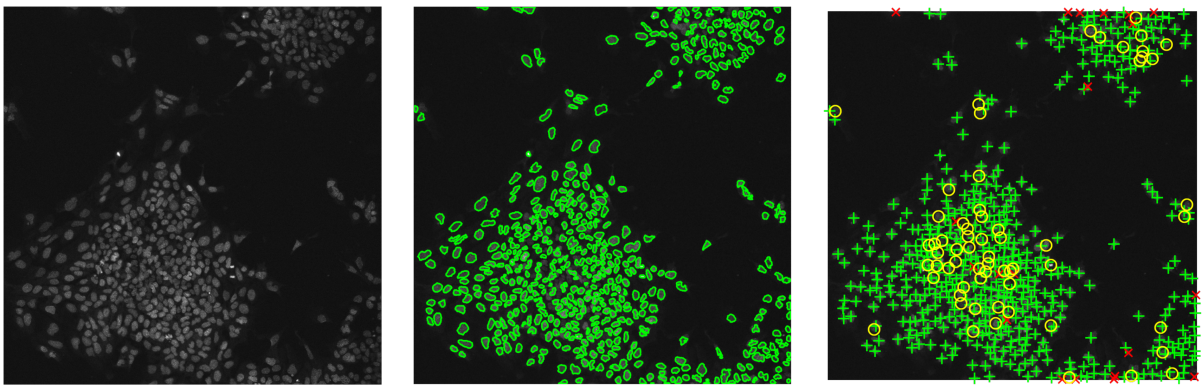
Original Selected regions Detections

Figure 3.9: *Example results of the (S+I) variant on the blastocysts dataset. The selected regions (*middle*) in this image should correspond to single instances of the blastocyst cells. In the detection image (*right*), correct detections are denoted with a green '+', false detections with a red 'x' and missed instances with a yellow 'o'.*



Original Selected regions Detections

Figure 3.10: *Example results of the (S+I) variant on the histopathology dataset. The selected regions (*middle*) in this image should correspond to single lymphocytes. In the detection image (*right*), correct detections are denoted with a green '+', false detections with a red 'x' and missed instances with a yellow 'o'.*



Original Selected regions Detections

Figure 3.11: *Example results of the (S+I) variant on the nuclei in fluorescence microscopy dataset. The selected regions (*middle*) in this image should correspond to single nuclei. In the detection image (*right*), correct detections are denoted with a green '+', false detections with a red 'x' and missed instances with a yellow 'o'.*

Method	Precision	Recall	F_1-score	$\mu_d \pm \sigma_d$	$\mu_n \pm \sigma_n$
Our method (S+I)	**85.89**±1.21	**89.90**±0.98	**87.85**±1.13	**1.68**±2.55	**2.90**±2.13
LIPSyM [84]	70.21	70.08	69.84	3.14±0.93	4.30±3.09
Bernadis et al. [20]	-	-	-	3.13±3.08	12.7±8.70
Kuse et al. [83]	65.23	69.99	67.29	3.04±3.40	14.01±4.40
Cheng et al.[39]	-	-	-	8.10±6.98	6.98±12.5
Graf et al. [63]	-	-	-	7.60±6.30	24.5±16.2
Panagiotakis et al. [117]	-	-	-	2.87±3.80	14.23±6.30

Table 3.1: *Results for the dataset of the ICPR 2010 Pattern Recognition in Histopathological Images contest [66]. Seven measures are reported: precision, recall and F1-score (when available), where higher numbers are better, and the four measures used in the evaluation of the ICPR contest, where lower numbers are better. The contest criteria consisted of the mean and standard deviation of two measurements: the Euclidean distance between detected dots and ground truth dots (***d***), and the absolute difference between the number of cells found and the ground truth number of cells (***n***).*

the inaccurate class labels brought by the high visual similarity between the lymphocytes (cells of interest in this task) and the breast cancer cells. Finally, Table 3.2 compares our results (S+I variant) on the phase contrast dataset to other recent methods for cell detection in microscopy images. In this case, our method is outperformed by a small margin (considering the size of the dataset) by [170], a cell detection and segmentation method based on correlation clustering.

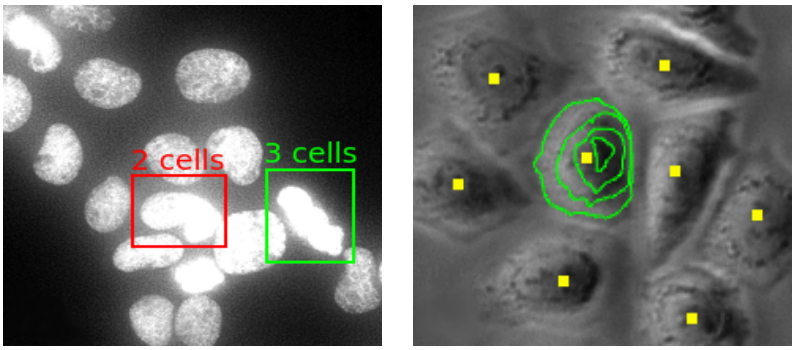
Table 3.2: *Results for the phase contrast dataset. Three measures are reported: precision, recall and F1-score (when available), where higher numbers are better. The evaluation of [94, 105, 170] was taken from [170]. The evaluation of [20] was kindly provided by the author, and it required the masking of the non-cell area as the method does not discriminate the detected regions.*

Method	Precision	Recall	F_1-score
Our method (S+I)	93.70 ± 0.20	91.94 ± 0.72	92.81 ± 0.35
Zhang et al. [170]	-	-	**95**
Logg et al. [94]	-	-	35
Mayer et al.[105]	-	-	32
Bernadis et al. [20]	85	84	85

3.6 Summary and Limitations

We have presented a method for object detection in microscopy images with application to cell detection that is able to maintain excellent performance across different scenarios, thus being robust to changes in image intensities, cell densities and cell sizes, while being specific to the structures of interest. The method uses extremal regions as detection candidates, and scores them with a learning-based measure. An in-built non-overlap constraint allows the method to perform well in the presence of cell clumping (without overlapping), and it was shown that the best performance is obtained when the learning process takes the non-overlap constraint into account.

One major limitation of the detection method is the requirement that there should exist at least a single candidate region (MSER) corresponding to each instance of the object of interest. However, such assumption can be easily broken in cases of object overlap (Figure 3.12a), which led to a poor performance in the synthetic, blastocysts and molecular imaging datasets, where instance overlap was considerable. Finally, a possible improvement is related to the appropriate use of the dot-annotations; as described so far, the model requires a heuristic procedure to define the "ground truth" set of candidate regions based on the dot-annotations (Figure 3.12b). Although heuristics can produce a good initial set of regions to learn from, it might not be optimal for the learning. Such extensions of the method are addressed in Chapter 4.



(a) (b)

Figure 3.12: *Limitations and extensions of the detection model in Chapter 3. a) Images can contain cell clusters with two, three or more cells, which cannot be easily differentiated due to low contrast caused by image saturation. b) Several extremal regions (shown for one cell in green) could be chosen to be the "ground truth" as they contain the annotation dot; however, choosing the best one for each cell during the training can improve the performance of the classifier.*

Chapter 4: Handling instance overlap in object detection

In Chapter 3, we described a learning-based method for detecting multiple instances of an object in microscopy images that was successful in cases of no or little object overlap. However, such a detection method assumes the existence of individual candidate regions for each of the instances of the object, which does not hold in many real applications and, in particular, in cases of strong object overlap. Therefore, the degree of object overlap plays a crucial role in the applicability of this class of detection method.

As discussed in Chapter 2, cases of strong object overlap and inter-occlusion, typical of high object density images, could be preferably handled through object density estimation methods. The analysis in this case is essentially reduced to texture matching between the test image and the training set, which may be feasible even when individual instances are not distinguishable. Nevertheless, for low object-density images with infrequent overlaps between them, detection methods may perform very well, while regression/density estimation methods can e.g. hallucinate small but non-zero object density/object count spread across the parts of images that do not in fact contain any objects. Furthermore, the localization of individual objects in the detection-based approaches facilitates more intricate analysis by revealing patterns of co-location, providing the possibility for shape and size estimation of individual instances, and allowing the linkage of individual detections through time for video analysis.

Many real applications require the processing algorithm to handle both the high and the low-density scenarios. Furthermore, the two cases may co-exist within the same image.

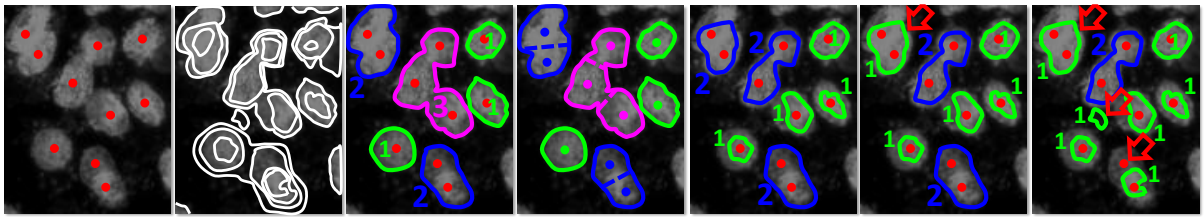
For instance, a microscopy image may contain both regions of low and high cell density (sometimes corresponding to different morphological parts or different tissues). Likewise, an image from a surveillance camera may contain multiple individual pedestrians but also few groups of people where the individuals are hard to segment from each other.

In such situations neither of the approaches mentioned above will perform optimally and this motivates the method we present in this chapter, which generalizes and builds upon our region-based detection method of Chapter 3. The main extension of the proposed method is its ability to parse the input image by detecting groups of objects of different integer sizes (with a "group" of size 1 being a particular case). Via training performed on a set of weakly-annotated[1] training images, the proposed method learns to choose different group sizes depending on the object density. Thus, similarly to local density estimation, it can avoid trying to discern individual objects when they are clumped together. Unlike local density estimation, however, the proposed method is able to enforce the fact that each group has an integer number of objects.

As in the base detection method of Chapter 3, the parsing process is based on an efficient and exact inference procedure that detects a set of non-overlapping extremal regions delivering a maximum to the parsing functional, and its extension is described in Section 4.2.

The learning, described in Section 4.3, is again performed in a structured SVM framework which optimizes the (convex upper bound on the) counting loss. We observe that such learning yields a desirable bias to prefer the most detailed explanation, e.g. to choose the groups of the smallest size whenever objects are discernible, as this strategy tends to provide the highest counting accuracy. Additionally, the learning is extended to model the ground truth based on dot-annotations in a better way; instead of selecting it from the regions that contain the expected number of dots, the ground truth configuration is mod-

[1] We consider dot-annotations to be "weak" in the sense that they do not capture the extent of the object as pixel-wise or bounding-box annotations, which is a compromise towards facilitating the work of the annotator. Nevertheless, in this context, "weak" annotations do not mean incomplete or image-level annotations.



a) input and ground truth | b) pool of nested regions | c) solution with loss=0 | d) post-proc. of (c) | e) solution with loss=0 | f) solution with loss=1 | g) solution with loss=3

Figure 4.1: *Given an input image (a) our model considers a pool of nested regions (b) and accomplishes detection by picking a non-overlapping subset of regions (c), where each region is assigned a label corresponding to the estimated number of objects (green=1, blue=2, purple=3). Such solution can be further refined to estimate individual object locations (d). The learning in our model is performed based on weak annotation (red dots) and is driven by an* instance count *loss. Solutions with zero loss (c and e) as well as non-zero loss (f and g) are shown. In the latter case, arrows indicate violations from the perfect correspondence between the solution and the ground truth dotting.*

eled as a latent variable also taken into account during the structural-SVM formulation, thus generalizing the approach described in Chapter 3.

In Section 4.5, we present a set of experiments with real and synthetic fluorescence microscopy images (datasets of Section 3.1), as well a surveillance data from the UCSD pedestrian dataset in order to show the generality of the approach. We observe that the proposed method achieves very good detection accuracies despite large amount of overlap, and very low effective resolution. For all datasets used for evaluation, the proposed method outperformed other detection methods, providing a considerable improvement over the baseline in Chapter 3, and it was found to be comparable with the methods that are trained to count (and do not perform detection). A conclusion is finally presented in Section 4.6.

4.1 Model overview

Firstly, we recall the model detailed in Chapter 3 and introduce its generalization. For an input image $\mathcal{I}$ containing multiple instances of an object class (some of which may be overlapping) we want to automatically detect the instances and provide an estimate of their location. We start by generating a pool of N *nested* regions (see Figure 3.3b for a

case where $N = 13$), such that for each pair of regions R_i and R_j in the pool, these regions are either nested (i.e. $R_i \subset R_j$ or $R_i \supset R_j$) or they do not overlap ($R_i \cap R_j = \emptyset$). In the simplest case, a pool can comprise extremal regions of the input image (i.e. connected components of the binary images $\mathcal{I} > \tau$ where τ is an arbitrary threshold).

Once the pool of nested regions is generated, each region is scored using a set of classifiers that evaluate the similarity of such region to each of D classes, where each class signifies the integer number of instances of the object that the region contains (i.e. a region of class d contains d instances). During the learning stage (detailed in Section 4.3), these classifiers are trained in a coordinated fashion within a structured output framework. Given the scores of the classifiers, an inference procedure (detailed in Section 4.2) selects a *non-overlapping* subset of regions. The inference also assigns each selected region in the subset a class label that indicates the number of objects that our system believes this region represents. The choice of the region subset and the class labels are driven by the optimization process that simply maximizes the total classifier score corresponding to selected regions and class labels subject to a non-overlap constraint.

4.2 Inference on the model

The inference on the tuple detection model is, again, a generalization of the inference procedure described in Section 3.3, but with the introduction of D object classes. Given a set of nested candidate regions, let $V_i(d)$ denote the classifier score of a region R_i for class d (the higher the score, the more this region looks like a typical region containing d object centroids). For notational simplicity, we also define $V_i(0) = 0$. We introduce the optimization variables $\mathbf{y} = \{y_i | i = 1 \dots N\}$, where $y_i = 0$ means that the region R_i is not selected, and $y_i = d \in 1 \dots D$ means that the region R_i is selected and assigned class d. We denote with $\mathcal{Y}$ the set of all $\mathbf{y}$ that meet the non-overlap constraint, i.e. such that $\forall i, j$: if $R_i \cap R_j \neq \emptyset$ then $y_i \cdot y_j = 0$. Then the inference is accomplished through the following constrained maximization:

$$F(\mathbf{y}) = \max_{\mathbf{y}\in\mathcal{Y}} \sum_{i=1}^{N} V_i(y_i). \tag{4.1}$$

This constrained maximization simply tries to maximize the cumulative score of all selected regions, and it can be solved as previously described in Section 3.3 after adding the posissible D classes into the pairwise tree-structured MRF (3.2) in the following way: $W_i(d,d) = 0$, $W_i(d,0) = V_i(d)$, $W_i(0,d{>}0) = -\infty$, and $W_i(d_1, d_2 \neq d_1) = -\infty$ as long as $d_2 > 0$.

Again, the optimization task is accomplished via tree-based dynamic programming [118] (the max-sum version of the algorithm).

4.3 Learning region classifiers

The model for the evaluation of the candidate regions can be trained on weakly annotated (dotted) images and does not require more detailed annotations (e.g. bounding boxes). Thus, we again assume that we are given a set of images annotated with dots, where each dot is placed inside each instance of the object. The learning is driven by an *instance count loss (IC-loss)* (4.2), denoted as L_{IC}, that penalizes all deviations from the one-to-one correspondences between annotation dots and the selected regions (Figure 4.1).

Suppose we have M training images $\mathcal{I}^j$ indexed by j. Let d_i^j now be the number of dots contained in the candidate region R_i^j, N^j be the total number of candidate regions in $\mathcal{I}^j$, and D^j be the total number of dots in $\mathcal{I}^j$. The *IC-loss* imposed by such annotation on each possible region labeling $\mathbf{y}$ is formulated as:

$$L(\mathbf{y}^j) = \sum_{i=1}^{N^j} [y_i^j > 0]\Delta(d_i^j, y_i^j) + D^j - \sum_{i=1}^{N^j} [y_i^j > 0] d_i^j. \tag{4.2}$$

Here, the first term penalizes the deviations between the assigned class label y_i^j of the selected regions and the true number d_i^j of dots inside of it. The penalty is determined by the function $\Delta(\cdot,\cdot)$, described in Section 4.3.1. The last two terms correspond to the total number of unmatched (uncovered) dots for the $\mathbf{y}^j$ configuration under the non-overlap

constraint, and thus penalize false negatives (missed detections).

Assuming that the properties of each region R_i^j (i.e. region appearance) in the pool of candidates are characterized by the *feature vector* $\mathbf{f}_i^j$, we set the classification scores to be linear functions of these feature vectors: $V_i^j(d) = (\mathbf{w}_d \cdot \mathbf{f}_i^j)$, where $\mathbf{w}_d$ is the parameter vector for the dth class, and has the same dimensionality as the feature vector. The aim of the learning is to find a vector $\mathbf{w} = \left[\mathbf{w}_1^T, \mathbf{w}_2^T, \ldots, \mathbf{w}_D^T\right]^T$ so that the inference (4.1) produces configurations with low IC-loss.

A simple approach for learning $\mathbf{w}$ is to train binary classifiers for each of the D classes, in a one-versus-rest fashion. However, such an approach ignores the inference process and the non-overlap constraint imposed by the inference. We therefore perform learning within a structured output learning framework; specifically, a structured SVM [152]. Thus, since the loss (4.2) is discontinuous w.r.t. $\mathbf{w}$ and hence cannot be optimized directly, a convex upper bound is optimized instead. The minimization objective on $\mathbf{w}$ can then be written as:

$$\min_w ||\mathbf{w}||^2 + \frac{C}{M}\sum_{j=1}^{M} \max_{\mathbf{y}^j \in \mathcal{Y}^j} \left(L_{IC}(\mathbf{y}^j) + \mathbf{w} \cdot \left(\Psi(f^j, \mathbf{y}^j) - \Psi(f^j, \bar{\mathbf{y}}^j)\right)\right), \tag{4.3}$$

where the first term is the regularization on $\mathbf{w}$, the second term is the upper bound on the training error, C is a constant that controls the trade-off between them, $\bar{\mathbf{y}}^j$ is some given "ground-truth" configuration (see later after (4.5)) with zero IC-loss, and $\Psi(\mathbf{f}^j, \mathbf{y}^j)$ is the *joint feature representation* defined as follows:

$$\Psi(\mathbf{f}^j, \mathbf{y}^j) = \left[\begin{array}{c} \sum_{i=1}^{N^j}[\mathbf{y}_i^j = 1]\mathbf{f}_i^j, \ldots, \sum_{i=1}^{N^j}[\mathbf{y}_i^j = D]\mathbf{f}_i^j \end{array}\right]^T. \tag{4.4}$$

The optimization objective (4.3) can be minimized with a cutting plane algorithm [152], for which an efficient way of computing the most violated constraint is required. Specifically, we need to compute the second term of equation (4.3) for a fixed $\mathbf{w}$ (*loss-augmented inference*). Fortunately, in our case the loss (4.2) decomposes in an appropriate way, and the loss-augmented inference corresponds to the following optimization (after removing

the terms independent from $\mathbf{y}^j$):

$$\max_{\mathbf{y}^j \in \mathcal{Y}^j} \sum_{i=1}^{N^j} [y_i^j > 0] \left(\Delta(d_i^j, y_i^j) - d_i^j \right) + \mathbf{w} \cdot \left(\Psi(f^j, \mathbf{y}^j) \right), \tag{4.5}$$

The maximization of (4.5) is then reduced to the optimization (4.1) with $V_i^j(y_i^j) = \mathbf{w}_{y_i^j} \cdot f_i^j + \Delta(d_i^j, y_i^j) - d_i^j$ and solved with the same dynamic programming inference.

Reestimating the "ground truth" configuration In the derivation above, the "ground truth" configuration $\bar{\mathbf{y}}$ was assumed given for each image; however, only dot-annotations are given at training time (not labeled regions), thus multiple "correct" (i.e. zero-loss) region configurations can be consistent with such annotation (Figure 4.1c,e). To handle this, we follow a conventional way [169] and add the "ground truth" configuration for each image into the optimization (4.3) as a latent variable $\mathbf{h}^j \in \mathcal{H}^j$ (where $\mathcal{H}^j$ denotes the set of all labelings with the zero IC-loss). The learning is reformulated as the following optimization:

$$\min_{\mathbf{w}, \mathbf{h}^j \in \mathcal{H}^j} \left\{ ||\mathbf{w}||^2 + \frac{C}{M} \sum_{j=1}^{M} \max_{\mathbf{y}^j \in \mathcal{Y}^j} \left(L(\mathbf{y}^j) + \mathbf{w} \cdot \Psi(f^j, \mathbf{y}^j) \right) - \frac{C}{M} \sum_{j=1}^{M} \mathbf{w} \cdot \Psi(f^j, \mathbf{h}^j) \right\}. \tag{4.6}$$

The new objective can then be optimized by alternation. This implies that we need to provide a way of imputing the latent variable such that the problem is reduced to the standard structured SVM in (4.3) for each iteration of the alternation algorithm. Specifically, at the beginning of iteration t for each training image j, we need to find $h^j \in \mathcal{H}^j$ that maximizes $\sum_{j=1}^{M} (\mathbf{w} \cdot \Psi(f^j, \mathbf{h}^j))$. To achieve this, we run the optimization (4.1) over $\mathcal{Y}^j$ but set $V_i^j(y_i^j) = \mathbf{w} \cdot \Psi(f^j, \mathbf{h}^j) + d_i^j \cdot v - [y_i^j \neq d_i^j] \cdot N^j v$, where v is a very large positive constant. This choice of V_i^j ensures that the maximum in (4.1) is attained for a zero-loss configuration from $\mathcal{H}$ and that the costs of all such configurations differ

from $\sum_{j=1}^{M} (\mathbf{w} \cdot \Psi(f^j, \mathbf{h}^j))$ by the same constant $N^j\, v$.

4.3.1 Penalization function for the IC-loss

In this subsection, we explore different choices for the penalization function within the IC-loss, one of which is then used in the experimental Section 4.5. The simplest choice for the penalization function $\Delta(d_i^j, y_i^j)$ is to directly measure the difference between the class y_i^j of a region and the number d_i^j of dots it contains as $\Delta^u(d_i^j, y_i^j) = |d_i^j - y_i^j|$. This penalization has the same behaviour regardless of the estimated class or the true number of dots inside the region. However, when considering the possibility of regions containing multiple objects, we should take into account the increasing intra-class variability (e.g. of region shape) for higher-order classes, which is consequently a more demanding learning task for the classifier. Also, consider assigning a class 7 to a region that contains 6 instances. For many tasks this is not as bad as assigning a class 3 to a region with 2 instances, thus it should not be penalized as heavily. To address such issues, we propose several variants of the penalization function, and their evaluation is detailed next.

$\Delta^u(d_i^j, y_i^j)$	$\lvert d_i^j - y_i^j \rvert$
$\Delta^{x^2}(d_i^j, y_i^j)$	$(d_i^j - y_i^j)^2$
$\Delta^s(d_i^j, y_i^j)$	$\lvert y_i^j - d_i^j \rvert / (d_i^j + 1)$
$\Delta^a(d_i^j, y_i^j)$	$\begin{cases} (y_i^j - d_i^j)/(d_i^j + 1), \text{ if } y_i^j \geq d_i^j \\ d_i^j - y_i^j, \text{ if } y_i^j \leq d_i^j \end{cases}$
$\Delta^g(D_i^j, y_i^j)$	$\begin{cases} (y_i^j - D_i^j)/(D_i^j + 1), \text{ if } y_i^j \geq D_i^j \\ D_i^j - y_i^j, \text{ if } y_i^j \leq D_i^j \end{cases}$, where $F_0^j = \sum_{P \in \mathcal{P}^j} \mathcal{N}(p; P, \sigma)$ and $D_i^j = \sum_{p \in R_i^j} F_0^j(p)$

Table 4.1: Variants of the penalization function $\Delta(\cdot, \cdot)$.

We evaluate the variants of the penalization function $\Delta(\cdot, \cdot)$ shown in Table 4.1. We first present simple variations where a region R_i^j has a cost equal to the absolute (Δ^u) or squared (Δ^{x^2}) difference between the class y_i^j assigned to R_i^j and the number d_i^j of dots it contains. We then consider the intuition that the penalization must compensate for the bias towards lower order classes created by the higher intraclass variability within

higher order classes. Therefore, in the variant Δ^s the difference between d_i^j and y_i^j is re-scaled in proportion to d_i^j, which effectively softens the penalization of errors in higher order classes. Δ^a re-scales penalties similar to Δ^s, but only in cases where there is a direct bias towards lower order classes; that is, when $y_i^j \geq d_i^j$. Finally, we introduce the variant Δ^g, with the same form as Δ^a but a key difference in how the true number of objects inside the region R_i^j is measured. As opposed to counting the number of dot-annotations inside a candidate region (d_i^j), we adopt the principle of "smoothed" dot-annotations of [90]. By placing Gaussian kernels centered on every dot-annotation of an image, we produce an object density map which allows us to evaluate candidate regions w.r.t. their coverage of objects. Let $\mathcal{P}^j$ be the set of dot-annotations in image $\mathcal{I}^j$. $F_0^j = \sum_{P \in \mathcal{P}^j} \mathcal{N}(p; P, \sigma)$ emulates a ground truth object density map such that integrating over any region in the image produces a non-negative real value indicative of the number of objects contained within such region. For notation simplicity we introduce $D_i^j = \sum_{p \in R_i^j} F_0^j(p)$ which represents the object density contained within the candidate region R_i^j (continuous analogous to d_i^j) and replaces d_i^j in Δ^g. Finally, we note that learning from the smoothed annotations would have a benefit similar to that of *jittering* the dot annotations for the purpose of training data augmentation.

The quantitative comparison between the variants of the penalization function is done over the synthetic fluorescence dataset (Figure 3.1a), for the splits of $N = 32$: five draws of 32 training images and 32 validations images (see Section 3.1 for details). The validation metric used in these experiments is the F_1-score score (i.e. the mean counting error reported also corresponds to the operating point of best detection accuracy and not that of lowest counting error). The results are shown in Table 4.2.

As expected, the unweighted penalization Δ^u results in the highest precision, but it tends to dismiss higher order classes, leading to a lower recall and higher counting error when compared to other variants of penalization function. The symmetrically re-scaled penalization Δ^s shows a more balanced performance by increasing the recall over Δ^u without much loss in precision. Finally, the asymmetric functions Δ^a and Δ^g achieve

	Precision	Recall	F_1-score	MCE
Δ^u	98.52 ± 0.06	87.62 ± 0.07	92.75 ± 0.04	19.00 ± 0.19
Δ^{x^2}	92.58 ± 1.26	89.07 ± 1.38	90.77 ± 0.14	12.65 ± 2.75
Δ^s	96.75 ± 0.03	90.19 ± 0.01	93.35 ± 0.01	12.06 ± 0.81
Δ^a	95.00 ± 0.96	91.38 ± 0.75	93.15 ± 0.13	7.98 ± 2.07
Δ^g	95.00 ± 0.07	91.97 ± 0.04	93.46 ± 0.01	7.31 ± 1.09

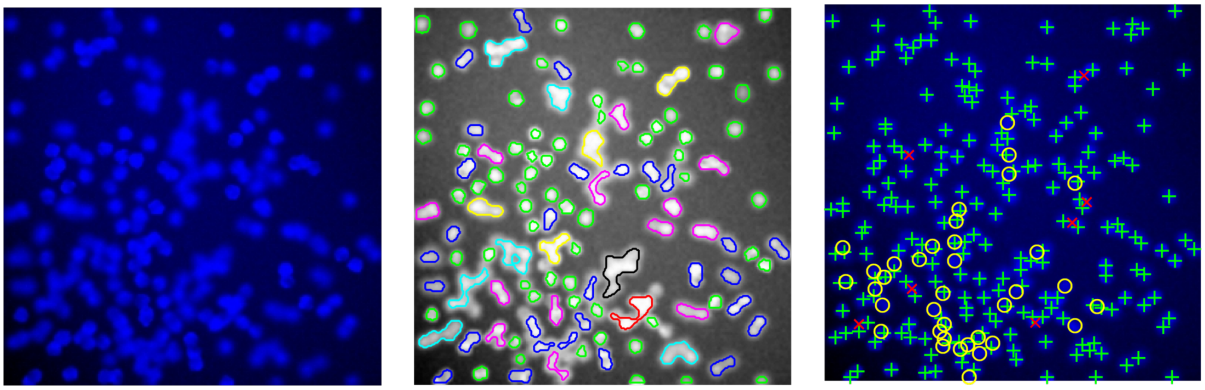
Table 4.2: Evaluation of the variants of the instance-count loss function for a detection-based (F_1-score) validation. *Penalizing errors concerned with higher order classes less than those with lower order classes results in higher recall and lower counting errors. See Table 4.1 and the text for the definitions of the functions.*

the best precision-recall balance, while providing a significant improvement in the mean counting error over all other variants. The difference in performance between Δ^a and Δ^g is minor. However, the qualitative examination of the results shows that the regions selected tend to better delineate objects of interest when using Δ^g. This is expected as Δ^g encourages the selection of regions that fully cover the objects of interest.

4.4 Implementation details

Post-processing for inference. Several potential applications and performance measures require the output of the method to be in the form of the sets of individual instances. We use a very simple post-processing in this case. For each selected region R_i we run k-means with $k = y_i$ on the image coordinates of all pixels in that region, thus obtaining an estimate for the set of centroids of individual objects. An example of this post-processing is shown in Figure 4.1(d).

Initialization and termination for learning. The initialization of $\mathbf{w}$ for the alternation-based maximization (4.6) is obtained by learning and concatenating a set of D binary classifiers $\mathbf{w}_1, \mathbf{w}_2, \cdots \mathbf{w}_D$ in a one-versus-rest fashion. The positive training examples for the binary classifier $\mathbf{w}_d$ consist of all regions in the training images that contain d dots. The alternations are stopped once the amount of change in the ground truth configuration with respect to the previous iteration $\frac{||\bar{\mathbf{y}}_t - \bar{\mathbf{y}}_{t-1}||}{M}$ falls below a pre-specified threshold ϵ.



Original Selected regions Detections

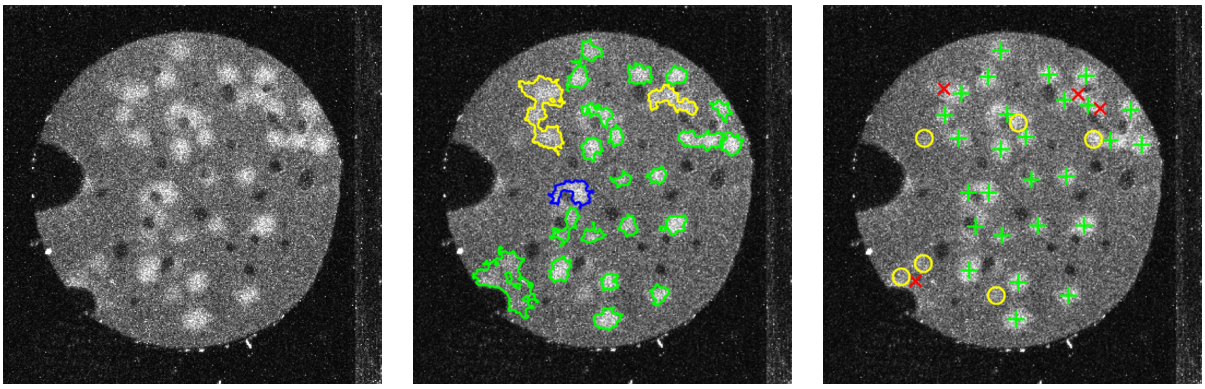
Figure 4.2: *Example results of the tuples detection on the synthetic dataset. The selected regions (*middle*) are colour-coded according to the number of instances they contain: green = 1, blue = 2, magenta = 3, yellow = 4, cyan = 5, black = 6 and red = 7. In the detection image (*right*), correct detections are denoted with a green '+', false detections with a red 'x' and missed instances with a yellow 'o'.*

4.5 Experiments

The main evaluation of the overlapping object detection method of this chapter, referred to as *tuples*, is done with the datasets and metrics presented in Section 3.1 and compared with the detection method of Chapter 3, referred to as *singletons*, as well as other published methods for detection and counting when available. Additionally, we aim to show the generality of the method by applying to the task of pedestrian detection in video surveillance, where the camera perspective leads to severe instance overlap. Example results are shown in Figure 4.8. The experiments of each group (microscopy and surveilleance cameras) are examined separately.

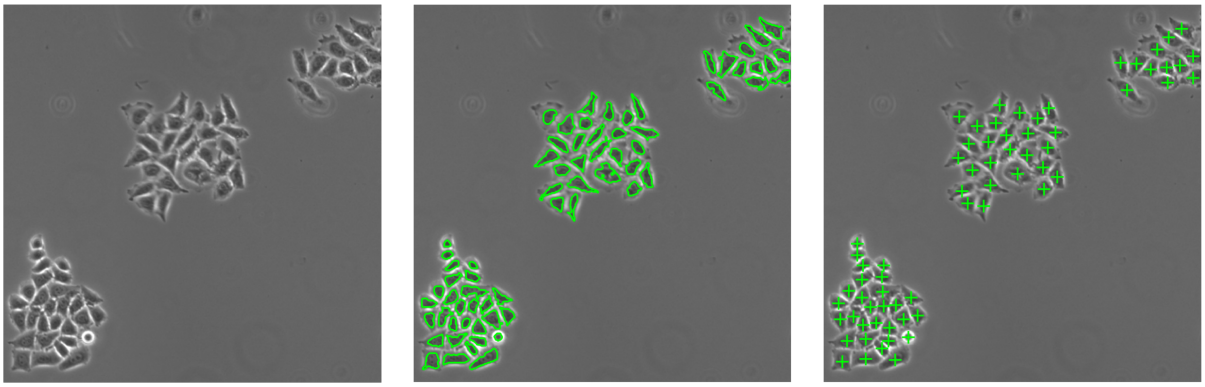
Microscopy datasets. For the experiments on the microscopy datasets (Section 3.1) we maintain the candidate region descriptors (visual features) as detailed in Section 3.5. The evaluation metrics for all datasets are shown in Table 4.3. Example results are shown in Figures 4.2, 4.3, 4.4, 4.5, 4.6 and 4.7.

The analysis of the metrics and example results shows that the extension of the detection method, as presented in this chapter, has a significantly positive effect for the datasets that present high-levels of instance overlap. A clear example is the performance



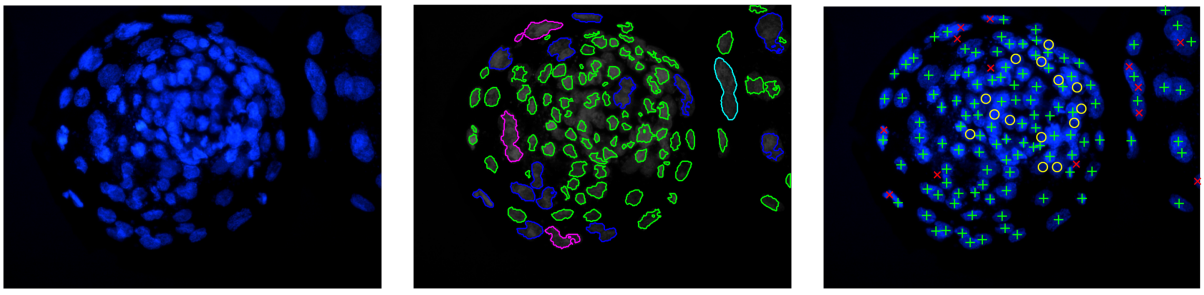
Original Selected regions Detections

Figure 4.3: *Example results of the (S+I) variant on the molecular dataset. The selected regions (*middle*) are colour-coded according to the number of instances they contain: green = 1, blue = 2, magenta = 3 and yellow = 4. In the detection image (*right*), correct detections are denoted with a green '+', false detections with a red 'x' and missed instances with a yellow 'o'.*



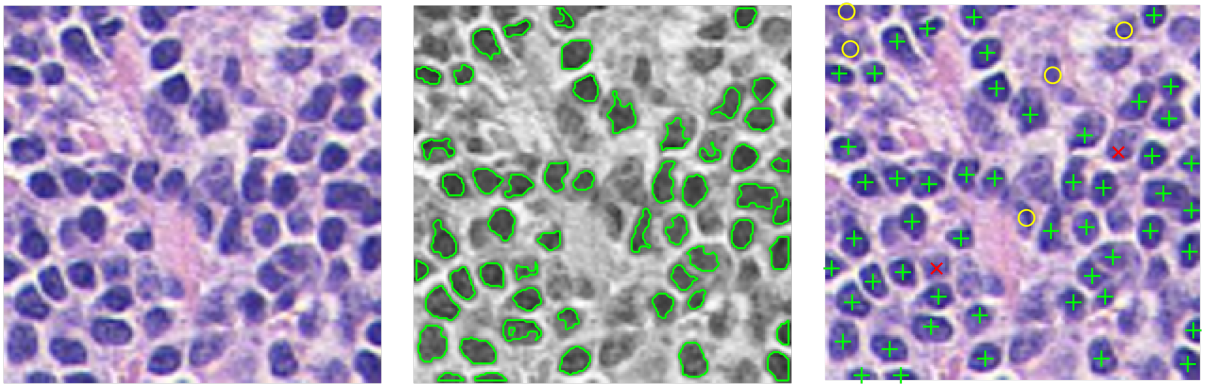
Original Selected regions Detections

Figure 4.4: *Example results of the tuples detection on the phase contrast dataset. The selected regions (*middle*) in this image correspond to single instances. In the detection image (*right*), correct detections are denoted with a green '+', false detections with a red 'x' and missed instances with a yellow 'o'.*



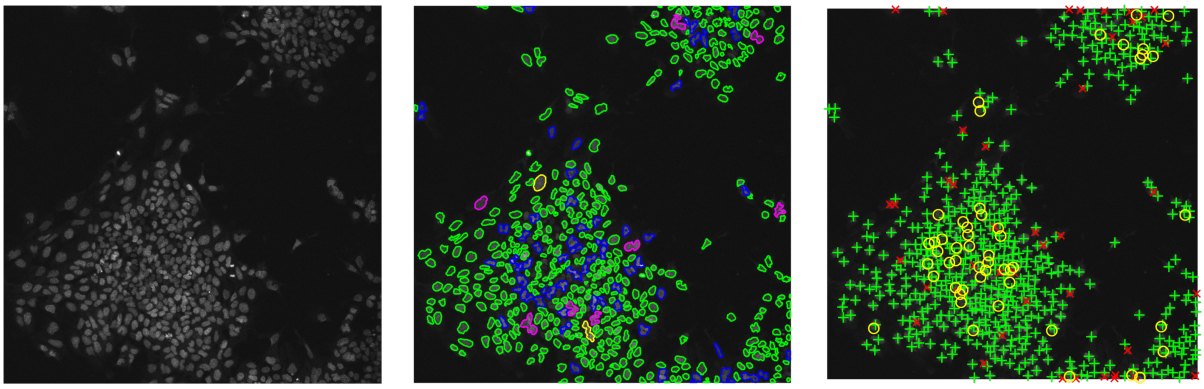
Original Selected regions Detections

Figure 4.5: *Example results of the tuples detection on the blastocysts dataset. The selected regions (*middle*) are colour-coded according to the number of instances they contain: green = 1, blue = 2 and magenta = 3. In the detection image (*right*), correct detections are denoted with a green '+', false detections with a red 'x' and missed instances with a yellow 'o'.*



Original Selected regions Detections

Figure 4.6: *Example results of the tuples detection on the histopathology dataset. The selected regions (*middle*) in this image correspond to single instances. In the detection image (*right*), correct detections are denoted with a green '+', false detections with a red 'x' and missed instances with a yellow 'o'.*



Original Selected regions Detections

Figure 4.7: *Example results of the tuples detection on the nuclei in fluorescence microscopy dataset. The selected regions (*middle*) are colour-coded according to the number of instances they contain: green = 1, blue = 2, magenta = 3 and yellow = 4. In the detection image (*right*), correct detections are denoted with a green '+', false detections with a red 'x' and missed instances with a yellow 'o'.*

on the synthetic dataset (Figure 4.2), where the singleton model would achieve a poor recall due to the inability to parse large clusters; on the other hand, the tuples model is even comparable to counting-based methods, which tend to be more appropriate for severe cases of object overlap as observed in the synthetic dataset. It is also noticeable that the performance of the tuples model is similar to that of the singleton model in cases of little object overlap, which is reflected on the fact that selected regions are mostly classified as singletons (e.g. Figure 4.4 and Figure 4.6).

Method	MCE	Precision	Recall	F_1-score
		Synthetic dataset		
Fiaschi *et al.* [53]	3.2 ± 0.1	-	-	-
Lempitsky & Zisserman [90]	3.5 ± 0.2	-	-	-
Barinova *et al.*[15]	6.0 ± 0.5	-	-	-
Singletons	51.2 ± 0.8	98.87 ± 1.52	72.07 ± 0.85	83.37 ± 1.20
Tuples	5.06 ± 0.2	95.00 ± 0.75	91.97 ± 0.43	93.46 ± 0.15
		Molecular dataset		
Singletons	15.59 ± 0.48	88.14 ± 1.75	41.19 ± 1.78	56.11 ± 1.51
Tuples	10.86 ± 0.40	81.65 ± 1.24	57.89 ± 1.27	67.73 ± 0.58
		Phase contrast dataset		
Correlation clustering [170]	-	-	-	95
Singletons	2.36 ± 0.67	93.70 ± 0.20	91.94 ± 0.72	92.81 ± 0.35
Tuples	3.84 ± 1.44	98.51 ± 1.16	95.76 ± 0.27	97.10 ± 0.27
		Blastocysts dataset		
Singletons	37.79 ± 1.11	97.51 ± 0.83	62.84 ± 2.49	76.39 ± 1.63
Tuples	13.89 ± 2.95	91.26 ± 2.73	77.00 ± 3.78	83.43 ± 1.56
		Histopathology dataset		
LIPSyM [84]	-	70.08	70.21	69.84
Singletons	3.7 ± 2.05	85.89 ± 1.21	89.90 ± 0.98	87.85 ± 1.13
Tuples	3.9 ± 2.65	84.09 ± 1.65	91.06 ± 1.5	87.40 ± 1.66
		Fluorescence dataset		
Singletons	46.82 ± 2.49	93.71 ± 0.23	81.74 ± 0.50	87.32 ± 0.19
Tuples	14.20 ± 4.59	90.14 ± 1.51	87.49 ± 1.23	88.78 ± 0.17

Table 4.3: Detection and counting accuracy for the microscopy datasets. *'Singletons' refer to the detection model presented in Chapter 3, whereas 'Tuples' refer to the model presented in this chapter. See Section 4.5 for details.*

Pedestrian dataset. We apply our method to detect and count pedestrians in the UCSD surveillance camera dataset [29]. It consists of 2000 frames (158×238 pixels) from a video surveilleance camera, annotated with a dot on each pedestrian and supplemented with an approximate depth image and the region of interest. The pedestrians frequently

occlude each other and are imaged at a very low resolution (the furthest pedestrians are just a few pixels tall). All this makes detection very hard for this dataset, and although a number of counting methods have been evaluated on it, to the best of our knowledge, we are the first to run detection algorithms.

As pedestrians can correspond to both dark and bright regions, we cannot use the extremal regions of the input images. Instead, to generate the tree of regions for this data, we computed the background image using a simple median filtering of a sparsely sampled set. For each frame, we then simply compute the absolute value of the difference with the background and look for extremal regions in this difference image. To reduce the number of candidate regions to a few hundreds, we applied a mild Gaussian smoothing to the difference image ($\sigma = 1$ pixel).

To describe each region, we have used (i) the histogram of visual words computed with tree codebooks as in [90], (ii) the area feature (as above), (iii) the histograms of intensities for the difference image, (iv) the histograms of Canny edge orientations as in [128], and (v) the nestedness feature (as above). All vectors were concatenated to obtain f_i^j.

We follow the protocol from [128] and split the data into four groups in order to assess accuracy, scalability and practicality. The first split, 'maximal', contains 128 frames out of a segment from the video, the splits 'upscale' and 'downscale' train on the most and least crowded frames respectively, and the 'minimal' split trains on only 10 frames. The counting results are shown in Table Table 4.4. In general, the proposed method outperforms the baseline Chapter 3. The counting accuracy of our detection method is comparable with the accuracy of methods that are trained to count and are not able to estimate the locations of individual pedestrians (even for singletons). For this dataset, we have observed that the method produced classes 1 to 5, indicating that discerning individual instances was harder than in the case of the real cell images.

In terms of the detection accuracy, the proposed method has also achieved an improvement over the baseline Chapter 3 (Table 4.5). This is due to the fact that the proposed

	'max'	'down'	'up'	'min'
Global count [80]	2.07	2.66	2.78	N/A
Segment+Count [128]	1.53	1.64	1.84	1.31
Density estim. [90]	1.70	1.28	1.59	2.02
Density estim. [53]	1.70	2.16	1.61	2.20
Singletons	2.55	2.25	2.93	2.86
Tuples	1.98	1.55	2.16	2.35

Table 4.4: Mean absolute errors for people counting in the surveillance video [29]. *The columns correspond to the four splits ('maximal','downscale','upscale','minimal'). Our detection method approaches the counting accuracy of the counting methods, while outperforming the baseline detection Chapter 3 in all splits.*

	'max'	'down'	'up'	'min'
Singletons	87.22	87.66	88.30	86.47
Tuples	89.53	89.99	89.21	86.64

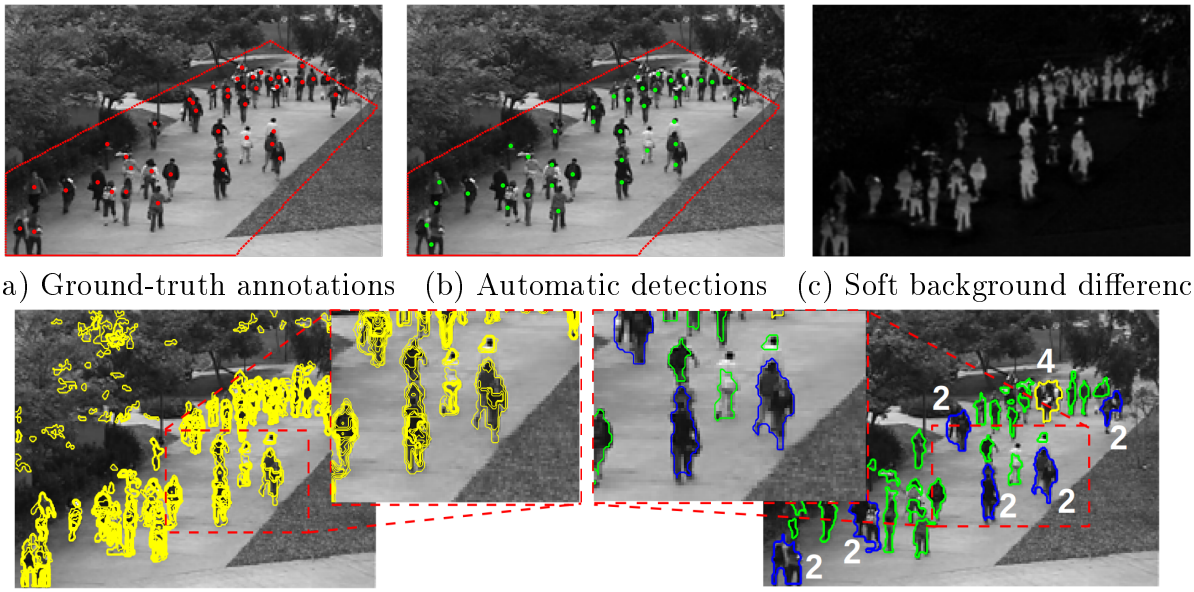
Table 4.5: Detection accuracy *in terms of 100*F_1 *score for the four splits of the UCSD pedestrian dataset. In this experiment, we varied the bias of the learned classifiers to generate recall-precision curves and picked the point with the highest F1-score on them. Generally, the proposed method resulted in higher optimal F1-score (and also reached the solutions with higher recall) compared to the baseline [9].*

method, while maintaining a precision similar to the baseline, is able to increase the recall as it has the capacity to handle overlapping objects.

Precision and recall curves. In Chapter 3 precision and recall curves were shown as an outcome of the method evaluation when detecting multiple individual instances of the objects (i.e singleton detection). In this chapter, however, precision and recalls curves have been obviated due to the fact that the method does not necessarily produce monotonic curves. The reason is that increasing a global bias for the score of the regions can cause the inference procedure to split high-order regions into several lower-order ones that do not necessarily preserve the recall (i.e. overall recall can decrease noticeably with a higher bias).

4.6 Summary and Limitations

We have presented a new model for object detection which generalizes the one of Chapter 3, and is particularly suitable for images with multiple overlapping object instances. Depending on the difficulty of the detection task, the model has the flexibility to choose



(a) Ground-truth annotations (b) Automatic detections (c) Soft background difference

(d) Candidate extremal regions (e) Selected regions

Figure 4.8: (best viewed in color) *Results for our methods on the UCSD pedestrian dataset. Due to the amount of overlap and low effective resolution, this dataset poses a big challenge for detection algorithms. Nonetheless, our method is able to produce accurate detection results (b) as compared to (a) the human annotations. Extremal regions are collected from (c) the soft background difference image (see text), and a portion of those regions is shown over the original image (d). The method selects non-overlapping regions (e) and estimates the number of instances of the object that the region contains, which allows the prediction of the location of the individual instances. Digits indicate the estimated number of instances inside the region, and green regions correspond to single objects.*

groups of variable sizes (including individual instances if the task is easy). The ability to pick the optimal level of granularity (i.e. to determine whether the task is "hard" or "easy") is seamlessly obtained during the learning of the model. The inference in the model remains computationally efficient, requiring only a few hundred classifier evaluations followed by tree-based dynamic programming.

The use of the model is particular attractive for biomedical images with some degree of object overlap, where it considerably outperforms the baseline of Chapter 3, which can only predict individual instances all the time. Thanks to the presented generalization of the region pool generation process, we could also apply the model to object detection in surveillance imagery, obtaining good detection accuracy despite low resolution.

One of the limitations of the proposed method appears when the instances become even denser than in the considered datasets and a higher number of classes is needed to parse such images. In this case, our structured output framework fragments the training data, so that higher-order classes effectively receive less training examples. A possible way of addressing such issue is using a learning framework that allows the sharing of information between the different classes (i.e. transfer learning). Another obvious possibility for improvement is a more sophisticated post-processing procedure (e.g. similar to [48]).

Finally, it is worth noting that all that is required of the candidate regions is that they are nested. Thus, although we have used extremal regions for candidates, they could instead be generated by hierarchical image segmentation, e.g. [6]. However, regardless of the method for candidate region detection, image information will play a crucial role for their generation, which might fail in cases of images with high levels of noise (e.g. weak-fluorescence images in the microscopy domain – Figure 3.1b), low contrast or images with highly inhomogeneous objects. In the case of the pedestrian dataset, the problem was circumvented by using temporal information. Nevertheless, in the case of the weak-fluorescence dataset, for example, this was not possible and the method did not achieve a decent performance due to the poor candidate regions. We propose in Chapter 5 a final

extension to the detection method that addresses the improvement of image conditions for the collection of better candidate regions.

Chapter 5: Optimizing microscopy images for object detection

In Chapter 4, we showed that the assumption of at least one candidate region per object instance could be relaxed by allowing regions to represent tuples of objects, but nevertheless, regular and smooth candidate regions play a crucial role in the performance of the resulting method. Collecting extremal regions as candidates for object detection from an intensity channel of microscopy images is often successful, as shown so far, but not optimal. For example, images with high levels of noise (e.g. weak-fluorescence images – Figure 3.1b), low contrast or images with objects of heterogeneous appearance can break the assumption that there exist extremal regions which can approximately represent each of the objects of interest or even a weaker assumption that extremal regions correspond to object groups. Nevertheless, for such cases, we show in this chapter that it is often possible to combine intensity channels and their modifications in order to obtain a new channel with extremal regions that are better suited for object detection. We refer to the height map defined over the generated 2D image channel as a *surface*. The computation of this surface, described in Section 5.1, can be done as a preprocessing step that is independent from other parts of the detection system. We then present in Section 5.2 experimental results that demonstrate the advantages of this preprocessing method, with a conclusion in Section 5.3.

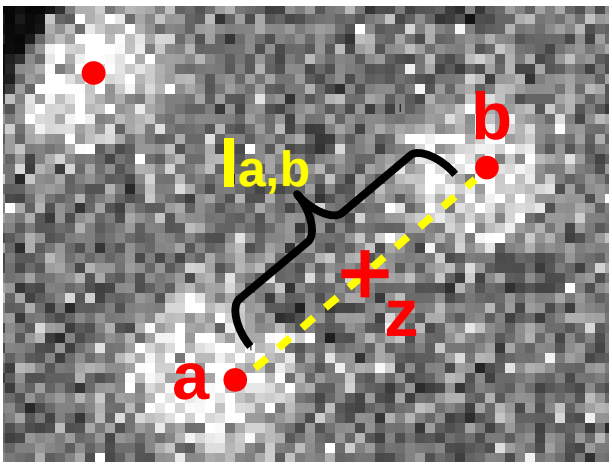


Figure 5.1: *The intuition behind the surface optimization (5.2) is that we want to collect extremal regions on a surface that is higher (by a margin) in the dot-annotations* **a** *and* **b** *than it is in a point* **z** *between them.* **z** *is a latent variable which indicates a location within the line* $l_{a,b}$ *connecting a pair of dot-annotations. See text for more detail.*

5.1 Surface computation

We propose to compute a surface optimized for extremal region collection in a supervised manner as a linear combination of feature channels, where a channel is a filtered version of the original image. That is, given a set of images $\mathcal{I}$, with their corresponding N feature channels $\mathcal{X}$, we aim to learn a weight vector α such that for any image $\mathcal{I}^j$, the surface can be computed as $S^j = \alpha_1 \cdot X_1^j + \alpha_2 \cdot X_2^j + ... + \alpha_N \cdot X_N^j$. In order to compute α, we design a cost function based on the following intuition. Assuming that we are focusing on bright blobs, an extremal region is a connected component of an image where all values inside of it are higher than all values on its boundary. Therefore, we want our surfaces to be *(i)* higher inside the objects of interest than between them, as well as *(ii)* smooth.

In order to enforce the condition *(i)*, we make use of the object localization supervision provided by the user in the form of dot annotations, which are also used to train the model described in Chapter 3 and Chapter 4, and are assumed to mostly lie within the objects of interest. Let **a** and **b** be the dot-annotations for two neighbouring instances of an object in our images, and **z** be a point between them whose selection is described below (see Figure 5.1). We want the surface S^j to be higher in **a** and **b** than it is in **z** by some margin. More generally, for every pair of neighbouring dot-annotations in $\mathcal{I}^j$, we

want $S^j(\mathbf{a}) \geq S^j(\mathbf{z}) + 1$ and $S^j(\mathbf{b}) \geq S^j(\mathbf{z}) + 1$. We build this constraint on the basis of pairs of neighbouring dots. More specifically, we consider each dot together with its closest neighbour (not necessarily reciprocal). Let the matrices $F(\mathbf{a})$, $F(\mathbf{b})$ and $F(\mathbf{z})$ denote respectively the values at the dot positions $\mathbf{a}$, $\mathbf{b}$ and $\mathbf{z}$ in each of the feature channels $\mathcal{X}$ associated to the images in $\mathcal{I}$ where they belong. For example, for a single image $\mathcal{I}^j$ with N feature channels and D dot-annotations $\mathbf{a}$, we define

$$F^j(\mathbf{a}) = \begin{pmatrix} X_1^j(\mathbf{a}_1) & X_1^j(\mathbf{a}_2) & \dots & X_1^j(\mathbf{a}_D) \\ X_2^j(\mathbf{a}_1) & X_2^j(\mathbf{a}_2) & & X_2^j(\mathbf{a}_D) \\ \vdots & & \ddots & \vdots \\ X_N^j(\mathbf{a}_1) & X_N^j(\mathbf{a}_2) & & X_N^j(\mathbf{a}_D) \end{pmatrix}$$

Therefore, $S^j(\mathbf{a}) = \alpha^T F^j(\mathbf{a})$ contains the values of the surface S^j at each dot $\mathbf{a}$. When using the entire training set $\mathcal{I}$, the matrices corresponding to each image are concatenated as $F(\mathbf{a}) = [F^1(\mathbf{a}), F^2(\mathbf{a}), \dots, F^J(\mathbf{a})]$. $F(\mathbf{a})$, $F(\mathbf{b})$ and $F(\mathbf{z})$ are used to easily compute the margin violations within the constraints of the optimization (5.2), where one slack variable $\xi_{a,b}$ is introduced for every pair $\mathbf{a}$ and $\mathbf{b}$ of dot-annotations. The goal of imposing this condition *(i)* can be also seen as an attempt to minimize the affinities along the direct path between $\mathbf{a}$ and $\mathbf{b}$, which resembles the method maximin affinity learning method of [153].

To enforce the smoothness condition *(ii)*, we simply attempt to down-weight "noisy" feature channels by measuring the standard deviation in the distribution of their Laplacian. For a single image $\mathcal{I}^j$ with N feature channels, we build the vector $\mathbf{L^j}$ containing the standard deviation of the Laplacian of each feature channel:

$$L^j = [\sigma(\triangle X_1^j), \sigma(\triangle X_2^j), \dots, \sigma(\triangle X_N^j)]^T. \tag{5.1}$$

For the entire training set $\mathcal{I}$, we compute a single vector $\mathbf{L}$ as the mean standard deviation of the corresponding feature channels. Finally, we find α through the minimization

$$\begin{aligned} \min_{\alpha,\xi} \quad & \alpha^T L + \lambda \sum_{\forall a,b \in \mathcal{D}} \xi_{a,b} \\ \text{s.t.} \quad & \alpha^T (F(a) - F(z)) \geq 1 + \xi, \\ & \alpha^T (F(b) - F(z)) \geq 1 + \xi, \\ & \xi \succeq 0, \alpha \succeq 0. \end{aligned} \tag{5.2}$$

The parameter λ controls the weights between the smoothness and margin violation terms in the cost function and is determined through cross-validation. In more detail, we compute on a validation set the number of margin violations for a set of values of λ. The notion of margin violation is the same as used in the optimization (5.2). We choose the λ with the lowest number of margin violations which is also within a pre-defined level of noise, measured through $\alpha^T L$ on the validation set.

Selection of z The variable $\mathbf{z}$ corresponds to the location between every pair of dot-annotations which would serve as reference for the optimization in (5.2). However, in contrast to the dot-annotations, the locations of $\mathbf{z}$ are unknown in advance. We choose to model $\mathbf{z}$ as latent variables, and thus, the optimization (5.2) is alternated with the imputation of $\mathbf{z}$. The latent variable is initialized as the set of middle points of line segments $l_{a,b}$ connecting $\mathbf{a}$ and $\mathbf{b}$ (Figure 5.1). For subsequent iterations, $\mathbf{z}$ is determined as $\min_z S^j(z), \forall z \in l_{a,b}$, that is during each imputation the line segment point with the lowest surface value is selected.

Implementation details. In all of our experiments, the feature channels $\mathcal{X}$ computed for the surface derivation in every image consist of *(i)* five scales of Gabor filter, each of which is the sum of the Gabor filters at different orientations, *(ii)* the original image blurred with eight different Gaussian kernels, and *(iii)* differences of the blurred images (difference of Gaussians). In case of color images, the luminosity channel of the *Lab* color space is used as the original image. Within the cross-validation of the hyperparameter λ of (5.2), the noise limit of the resulting surface is set empirically to 0.1. The time required for the surface learning varies depending on the number of data points, but in

our experiments is in the range of minutes. At testing time, generating the surface given the weight vector takes under a second as it only implies computing the global features and combining them linearly.

5.1.1 Validation experiments

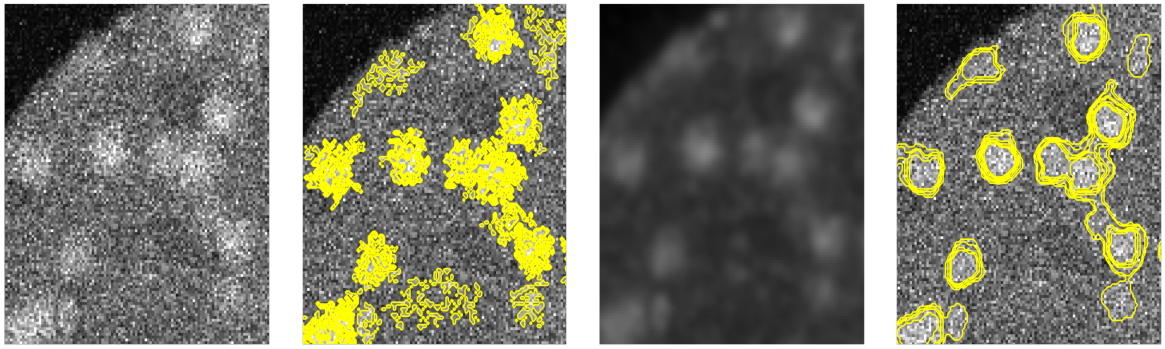
In order to demonstrate the usefulness of the surface optimization, we assess the performance of the model on the weak-fluorescence molecular dataset (Figure 3.1b) with and without this pre-processing step.

Qualitatively, it can be seen (Figure 5.2) that the surface optimization procedure has two positive effects: first, due to the smoothness enforced on the surface (Figure 5.2c), the pool of candidate regions (Figure 5.2d) is both smaller and with higher quality (i.e. regions better approximate the boundaries of the objects) than the one obtained from the original image (Figure 5.2b); secondly, due to the margin imposed on the surface computation, the contrast of the objects is enhanced leading to a higher recall in the object detection. Quantitatively, Table 5.1, the surface computation on the molecular dataset leads to higher detection accuracy and lower computation time per image due to the reduced number of candidate regions.

We conduct a similar validation experiments on the synthetic dataset in order to assess the behaviour of the surface crafting in a case where the main challenge is the severe instance overlap. We found the surface crafting can produce a surface with an overall negative effect in the case where the detection method is able to parse groups of objects (i.e. tuples detection). The reason is that, although the surface learning has managed to correctly break some of the clusters, in the cases of high-order clusters, these are broken into parts that resemble individual instances, but with less instances than the original cluster (See Figure 5.3). Therefore, the recall of the detection method is reduced (e.g. a cluster with 7 instances can be broken into 4 parts that resemble individual instances, thus causing the method to miss 3 instances). Nevertheless, we argue that the existence of such high-order clusters with heavily overlapping (and indistinguishable) instances is

	Precision	Recall	F_1-score	MCE
No surface optimization	81.65 ± 1.24	57.89 ± 1.27	67.79 ± 0.58	10.98 ± 0.34
Surface optimization	80.01 ± 3.62	75.09 ± 2.17	77.43 ± 1.98	7.13 ± 0.23

Table 5.1: Evaluation of the effect produced by the computation of candidate regions on an optimized surface. *The evaluation is done on the molecular dataset (Figure 3.1b).*



(a) Original (b) Region pool on (a) (c) Surface (d) Region pool on (c)

Figure 5.2: *The effect of surface optimization for the computation of candidate regions. When dealing with highly noisy images such as (a), the resulting pool of extremal candidate regions might not be appropriate (b). Through the computation of an optimized surface (c) for the collection of extremal regions, the pool of candidate regions (d) can be improved significantly. In this particular example, the surface (c) optimization has selected to keep and combine only four of the feature channels available: two of Gabor filters (two different scales), a channel of difference of Gaussians, and a channel of Gaussian smoothing.*

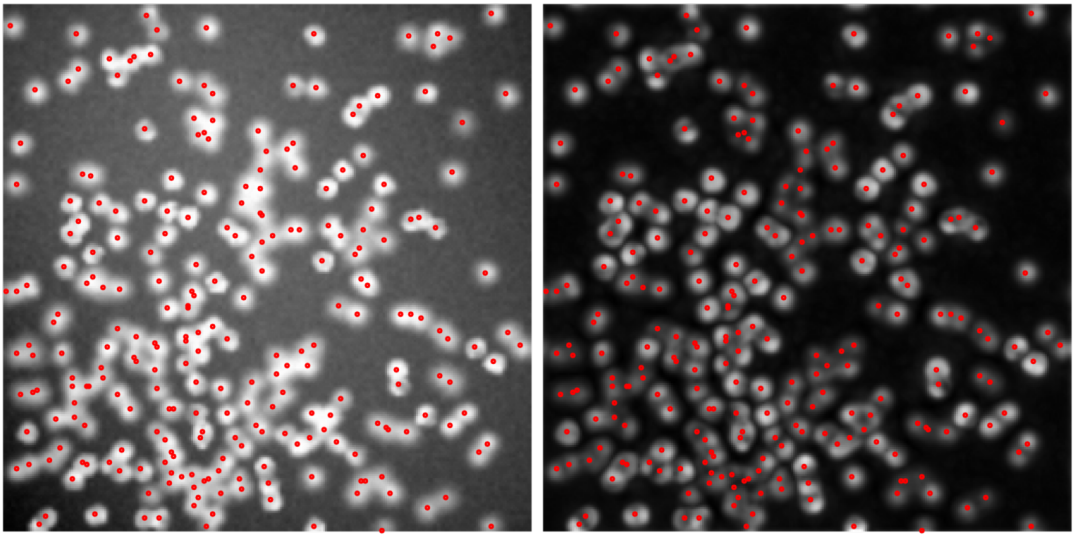
an artifact created by the synthetic nature of this dataset and we did not encounter such extreme cases in real microscopy images.

As a result of the possible failure, we use the surface only in cases where it is required, which can be decided through cross-validation.

5.2 Experiments

We now test the effect of the surface crafting preprocessing step on the datasets and with the metrics of Section 3.1. For all experiments, we maintain the candidate region descriptors (visual features) as detailed in Section 3.5. The evaluation on all datasets is shown in Table 5.2, and example results for the cases where the surface has been used are shown in Figures 4.3, 4.5 and 4.7.

Within our experiments, the molecular dataset shows the greatest benefit of the sur-



Original and annotations Surface and annotations

Figure 5.3: *Failure example of the surface learning on the synthetic datasets. Even though the surface learning has managed to correctly break some of the clusters, in the cases of severe overlap, clusters are broken into parts that resemble individual instances, but with less instances than the original cluster. Therefore, the recall of the detection method is reduced.*

face optimization for extremal region collection. The latter was observed in both the single and multiple class versions of our system, and it is an expected result considering the intuition shown in Figure 5.2 of the refinement of the candidate region pool. The blastocysts and fluorescence dataset were found to also benefit from the image enhancement brought by the surface crafting, showing moderate improvements in all the evaluation metrics.

With relatively limited cell overlap and mostly well-defined cell boundaries, experiments on the phase contrast microscopy and histopathology datasets showed no benefit from the surface computation component of our system, which would be therefore discarded during cross-validation

Finally, as seen detailed in the validation experiments of Section 5.1.1, the performance of the tuples models on the synthetic dataset was considerably harmed by the surface crafting.

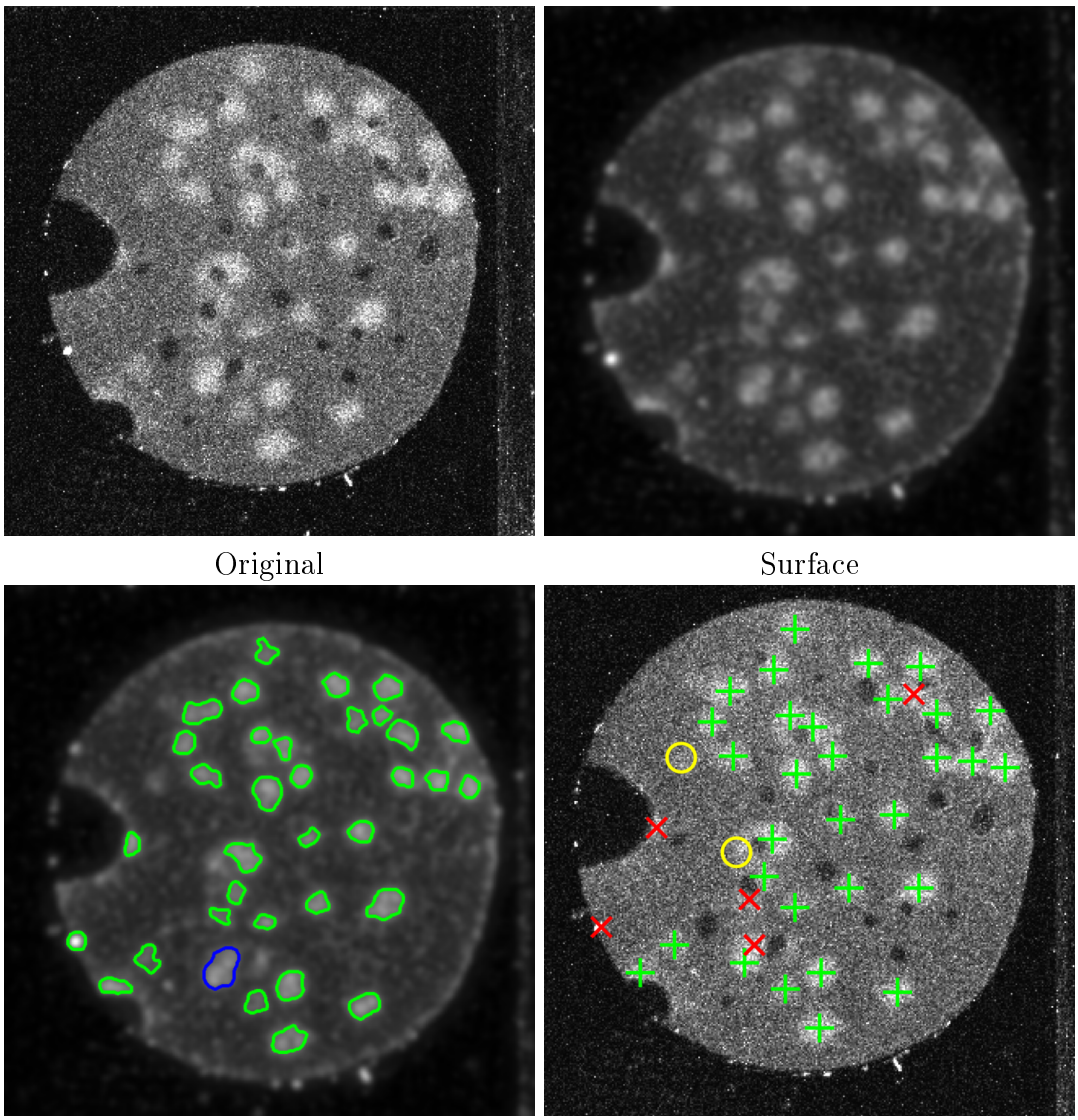


Figure 5.4: *Example detection result in the dataset of molecular imaging with weak-fluorescence. Selected regions (*bottom left*) are colour-coded according to the number of instances they contain: green = 1 and blue = 2. In the detection image (*bottom right*), correct detections are denoted with a green '+', false detections with a red 'x' and missed instances with a yellow 'o'.*

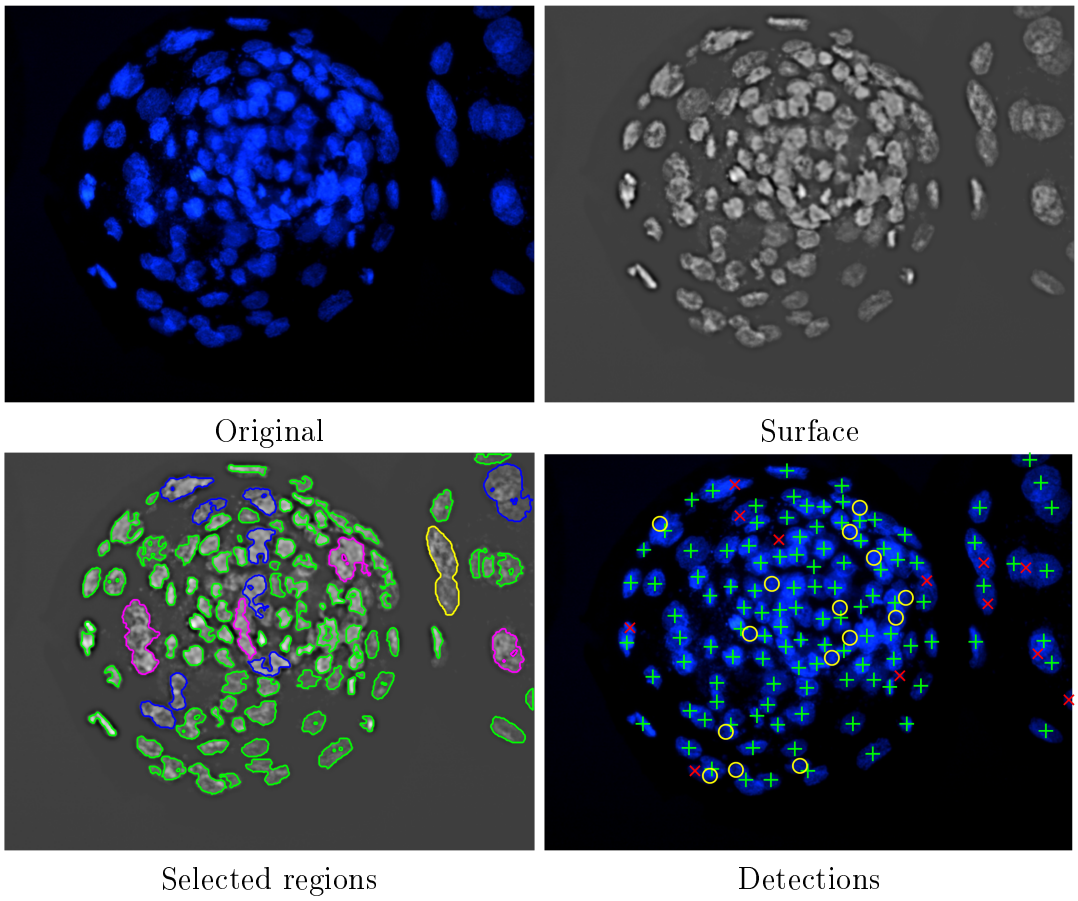


Figure 5.5: *Example detection result in the blastocyst dataset. Selected regions (*bottom left*) are colour-coded according to the number of instances they contain: green = 1, blue = 2, magenta = 3 and yellow = 4. In the detection image (*bottom right*), Correct detections are denoted with a green '+', false detections with a red 'x' and missed instances with a yellow 'o'.*

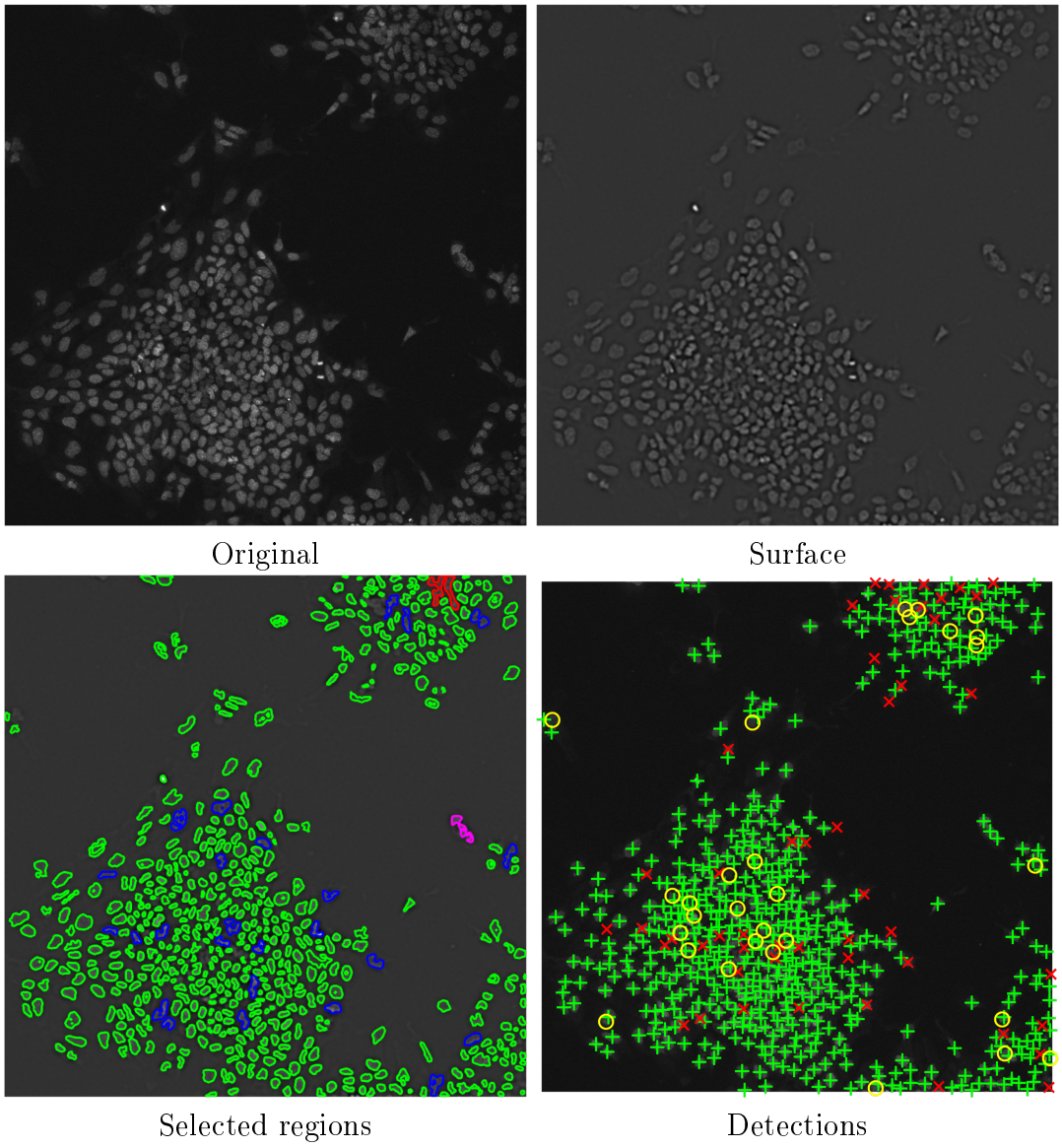


Figure 5.6: *Example of cell nuclei detection in fluorescence microscopy. Selected regions (*bottom left*) are colour-coded according to the number of instances they contain: green = 1, blue = 2, magenta = 3 and red = 7. In the detection image (*bottom right*), correct detections are denoted with a green '+', false detections with a red 'x' and missed instances with a yellow 'o'.*

Method	MCE	Precision	Recall	F_1-score
		Synthetic dataset		
Fiaschi *et al.* [53]	3.2 ± 0.1	-	-	-
Lempitsky & Zisserman [90]	3.5 ± 0.2	-	-	-
Barinova *et al.*[15]	6.0 ± 0.5	-	-	-
Singletons	51.2 ± 0.8	98.87 ± 1.52	72.07 ± 0.85	83.37 ± 1.20
Tuples	5.06 ± 0.2	95.00 ± 0.75	91.97 ± 0.43	93.46 ± 0.15
Tuples + surface	-	-	-	-
		Molecular dataset		
Singletons	15.59 ± 0.48	88.14 ± 1.75	41.19 ± 1.78	56.11 ± 1.51
Singletons + surface	6.88 ± 0.50	84.01 ± 2.59	69.75 ± 1.54	76.20 ± 1.61
Tuples	10.86 ± 0.40	81.65 ± 1.24	57.89 ± 1.27	67.73 ± 0.58
Tuples + surface	7.12 ± 0.36	80.01 ± 3.62	75.09 ± 2.17	77.43 ± 1.98
		Phase contrast dataset		
Correlation clustering [170]	-	-	-	95
Singletons	2.36 ± 0.67	93.70 ± 0.20	91.94 ± 0.72	92.81 ± 0.35
Singletons + surface	5.55 ± 2.70	91.20 ± 1.31	87.21 ± 3.53	89.13 ± 2.12
Tuples	3.84 ± 1.44	98.51 ± 1.16	95.76 ± 0.27	97.10 ± 0.27
		Blastocysts dataset		
Singletons	37.79 ± 1.11	97.51 ± 0.83	62.84 ± 2.49	76.39 ± 1.63
Singletons + surface	25.35 ± 2.03	94.59 ± 0.29	72.87 ± 1.29	82.31 ± 0.78
Tuples	13.89 ± 2.95	91.26 ± 2.73	77.00 ± 3.78	83.43 ± 1.56
Tuples + surface	9.24 ± 1.52	90.47 ± 1.00	81.77 ± 1.18	85.90 ± 0.94
		Histopathology dataset		
LIPSyM [84]	-	70.08	70.21	69.84
Singletons	3.7 ± 2.05	85.89 ± 1.21	89.90 ± 0.98	87.85 ± 1.13
Singletons + surface	4.1 ± 2.97	84.12 ± 1.20	87.46 ± 1.72	85.73 ± 1.88
Tuples	3.9 ± 2.65	84.09 ± 1.65	91.06 ± 1.5	87.40 ± 1.66
		Fluorescence dataset		
Singletons	46.82 ± 2.49	93.71 ± 0.23	81.74 ± 0.50	87.32 ± 0.19
Singletons + surface	16.90 ± 1.83	89.57 ± 1.10	88.48 ± 0.83	89.01 ± 0.19
Tuples	14.20 ± 4.59	90.14 ± 1.51	87.49 ± 1.23	88.78 ± 0.17
Tuples + surface	20.42 ± 4.10	87.12 ± 1.17	91.10 ± 0.75	89.05 ± 0.29

Table 5.2: Detection and counting accuracy for the microscopy datasets. *'Singletons' refer to the detection model presented in Chapter 3, 'Tuples' refer to the model presented in Chapter 4, and 'Surface' refers to the preprocessing step of image enhancing presented in this chapter. See Section 4.5 for details.*

5.3 Summary and Limitations

We have presented a pre-processing step which takes the input images and generates a smooth and contrast-enhanced surface that is optimized for the collection of extremal regions as object detection candidates. Such a step can be used prior to the detection method of Chapter 4 in order to handle particularly challenging scenarios such as detec-

tion on noisy microscopy imaging modalities. This is due to the fact that the quality of the pool of candidate regions is a key issue as good delineation of the objects of interest seems to facilitate learning good features for the classification stage.

We found this generated surface to be helpful in most of our experiments with overlapping instances, not helpful in the cases of mostly non-overlapping instances, and harmful in the case of the synthetic dataset which contains large clusters of extremely overlapping instances. Variants of the surface could be produced in different ways that could be more appropriate for cases where the objects of interest have a much more complex appearance such as in human detection. One example of an alternative surface would be to compute a pixel-wise probability map of individual object detections.

We note that the surface is not restricted to the usage of extremal regions as candidates, and it can benefit other candidate regions if the intentions is to produce a nested set such that they result in tree-structured graphical models. For example, recursive spectral clustering or superpixel merging.

Chapter 6: Object density estimation for instance counting

In Chapter 4 we discussed an object detection method that blurs the boundaries between classical detection and object counting for the case of object overlap. Nevertheless, as explained in Chapter 2, there are scenarios where detection is not at all possible due to challenges such as extreme overlap or very low effective spatial resolution.

In this chapter we begin the exploration of counting methods through density estimation, which are introduced in detail in Section 6.1 along with the counting framework of Lempitsky and Zisserman [90], used as a reference learning method for density estimation. We then propose, in Section 6.2, a simplified approach for supervised density learning based on ridge regression (i.e. simple linear algebra operations), and we show that this approach achieves similar counting accuracy to the constraint-generation based learning in [90] (using the same features), while being dramatically faster to train. We continue to explore, in Section 6.3, how to constrain the learning of the density estimation in order to produce results that are "smooth" in time for time-lapse microscopy. Finally, we show the application of density estimation to a real tasks in the area of radiation biology (Section 6.4). A summary and discussion of limitations are presented in Section 6.5.

6.1 Density estimation for object counting

We firstly describe in detail the object density estimation approach for object counting introduced in Section 2.3, with emphasis on the method of Lempitsky and Zisserman [90], which we use in the experimental sections of this chapter.

The task of density estimation consists in learning the mapping $\mathcal{F} : \mathcal{X} \mapsto \mathcal{Y}$ from local image features to a ground truth object density, where the ground truth object density is generally defined ([53, 90]) from the set of user dot-annotations $\mathcal{P}$ as the sum of delta functions centred on each of the annotation dots:

$$F^0(p) = \sum_{p' \in \mathcal{P}} \delta(p - p') \ . \tag{6.1}$$

Here, p is the (x, y) position of a pixel.

In the case of [90], the mapping is defined by a learned vector w such that for any image $i \in \mathcal{I}$, the object density can be calculated for any pixel p as a dot product between w and the pixel encoding vector x_p^i. That is, $F_i(p|w) = w^T \cdot x_p^i$, where F_i is the estimated cell density map in image i. Once F_i^0 is obtained, w is estimated in a regularized empirical risk minimization framework [21, 156]

$$w = \min_w \left(w^t \cdot w + \lambda \sum_{i=1}^{N} \mathcal{D}(F_i^0(.), F_i(.|w)) \right) \tag{6.2}$$

where the first term is the regularization on w, $\mathcal{D}$ is a distance between the ground truth and the estimated density, and λ controls the trade-off between the regularization and the training error. $\mathcal{D}$ is taken to be the square of the $MESA$ distance, defined as:

$$\mathcal{D}_{MESA}(F_1, F_2) = \max_{B \in \mathcal{B}} \left| \sum_{p \in B} F_1(p) - \sum_{p \in B} F_2(p) \right| \tag{6.3}$$

where F_1 and F_2 are the density maps being compared, $\mathcal{B}$ is the set of all possible rectangular sub-windows in them, and p indexes the pixels within a box $B \in \mathcal{B}$.

6.2 Density estimation through ridge-regression

We now introduce an alternative to the reference method of Lempistky and Zisserman [90] for learning an object density estimator. Our method here is similar to (and, arguably, simpler than) the density estimation method of Fiaschi *et al.* [53]. Most importantly,

compared to [90], this new approach reduces the training time from several dozens of seconds to a few seconds (for heavily annotated images).

The idea is to use the simple ridge-regression to learn the mapping $\mathcal{F} : \mathcal{X} \mapsto \mathcal{Y}$ from local image features to a ground truth object density. Let us assume that each pixel p in a training image $\mathcal{I}$ is represented with a sparse vector $x_p \in R^k$ from a learned codebook. At the same time, each pixel is associated with a real-valued ground truth object density $y_p \in R$ according to $F^0(p)$ in (6.1). The ridge regression finds a k-dimensional vector of coefficients w that minimizes the following objective:

$$||Xw - Y||^2 + \lambda||w||^2 \rightarrow \min_w \tag{6.4}$$

Here, X is the matrix of predictors (feature vectors) with each row containing x_p, Y is a vector of corresponding density values from the ground truth density F^0 and λ controls the balance between prediction error and regularization of w. Although computationally simple and efficient, the fitting procedure (6.4) tries to match the ground truth density values exactly in every pixel, which is unnecessary and leads to severe overfitting. Instead, as was argued in [90], the estimated densities Xw should match the ground truth densities Y when integrated over extended image regions (i.e. we do not care about very local deviations between Xw and Y as long as these deviations are unbiased and sum approximately to zero over extended regions).

Based on this motivation, [90] replaces the L2-distance between Xw and Y in (6.4) by the so called MESA distance which looks at integrals over all possible box regions. While this change of distance dramatically improves the generalization, the learning with MESA distance is costly as it employs constraint generation and has to solve a large number of quadratic programs. Here, we propose another, much simpler, alternative for the L2-distance in (6.4). Namely, we minimize a smoothed version of the objective by convolving the difference between the ground truth density and the estimated density with a Gaussian kernel:

$$||G * (Xw - Y)||^2 + \lambda||w||^2 \rightarrow \min_w \tag{6.5}$$

Here, $G*$ denotes (with a slight abuse of notation) the Gaussian smoothing over the image plane (i.e. the column vector has to be reshaped back to the image dimensions and smoothed spatially). Typically, we use a sufficiently large isotropic covariance to ensure that the local unbiased deviations between Xw and Y are smoothed (so that "excesses" and "deficits" cancel each other). We found that the performance of the estimated w on the test set is not sensitive to large variation in the covariance (as long as the covariance parameter σ within G is greater than say half a typical object diameter). The smoothed objective (6.5) is also better conditioned than (6.4) due to the contextual information brought by the spatial smoothing of the visual features, especially when these are one-hot encodings over a visual dictionary as used later on. Therefore, such smoothing is crucial for good performance of the counting.

Because of the linearity of the convolution, we can rewrite (6.5) as:

$$||(G * X)w - G * Y)||^2 + \lambda||w||^2 \rightarrow \min_w , \tag{6.6}$$

where $(G * X)$ denotes Gaussian smoothing applied to each column of X.

Importantly, (6.6) can be regarded as ridge regression between the smoothed version of the feature maps $G * X$ and the smoothed ground truth density. The latter can be seen as the sum of Gaussian kernels centered on the user annotations:

$$F^0(p) = \sum_{p' \in \mathcal{P}} \mathcal{N}(p', \sigma) \tag{6.7}$$

Similarly, the smoothed (but still sparse) matrix of predictors $G * X$ can be obtained by convolving independently each dimension of the feature vectors (i.e. each column of X), that is, spatially blurring each of the feature channels.

Using the vertically concatenated smoothed maps $X_s = [G * X]$ and $Y_s = [G * Y]$

respectively, w can be expressed using a standard ridge regression solution formula:

$$w = (X_s^T X_s + \lambda I)^{-1} X_s^T Y_s \,, \tag{6.8}$$

where I denotes the identity matrix and λ is a regularization parameter.

Finally, for a non-annotated image $\mathcal{I}$, the density value at pixel p can be obtained using the estimated w through a simple linear transform of the non-smoothed feature vectors x_p (i.e. in the same way as in [90]):

$$F(p) = w^T x_p \tag{6.9}$$

It can be seen that learning the mapping vector w only involves simple matrix operations (mostly on sparse matrices) and Gaussian smoothing. Thus, w can be learned on-the-fly and with a low memory footprint.

We have found that the generalization of the learning is improved slightly if the non-negativity of the estimated w is enforced (which for non-negative x_p results in physically meaningful non-negative object densities at test time). Since it is computationally more expensive to include a non-negativity constraint within ridge regression in a principled manner (compared to the closed-form solution provided by unconstrained ridge regression), we use a simple trick of iteratively re-running ridge regression, while clipping the components of w having negative values to zero after each iteration.

6.2.1 Experimental validation of ridge regression counting

We now evaluate the counting method based on ridge regression and determine how it compares to previous work in the area. In particular, we compare to the MESA-distance approach [90] (using the same features, the same datasets, and the same experimental protocols as in [90]).

Table 6.1 shows the experimental results on the USCD pedestrian dataset [29] (also used in Chapter 4), which consists of 2000 frames of dot-annotated pedestrians from a

	'max' [28.25]	'down' [24.35]	'up' [29.68]	'min' [28.25]
Global count [80]	2.07	2.66	2.78	N/A
Segment+Count [128]	1.53	1.64	1.84	1.31
Density estimation (MESA) [90] ⋆	1.70	1.28	1.59	2.02
Density estimation (RF) [53]	1.70	2.16	1.61	2.20
Density estimation (proposed) ⋆	1.24	1.31	1.69	1.49

Table 6.1: Mean absolute errors for people counting in the UCSD surveillance camera video [29]. *The columns correspond to four training/testing splits of the data ('maximal','downscale','upscale','minimal') proposed in [128]. The average number of people per image in each of the testing sets is shown in brackets on the top row. ⋆ indicates the methods tested with the same set of features. The proposed method (bottom line) matches on average the performance of the previous best method for the same features.*

	N = 1	N = 2	N = 4	N = 8	N = 16	N = 32
Img-level ridge reg. [90] ⋆	67.3 ± 25.2	37.7 ± 14.0	16.7 ± 3.1	8.8 ± 1.5	6.4 ± 0.7	5.9 ± 0.5
Dens. estim. (MESA) [90] ⋆	9.5 ± 6.1	6.3 ± 1.2	4.9 ± 0.6	4.9 ± 0.7	3.8 ± 0.2	3.5 ± 0.2
Dens. estim. (RF) [53]	N/A	4.8 ± 1.5	3.8 ± 0.7	3.4 ± 0.1	N/A	3.2 ± 0.1
Dens. estim. (no smooth) ⋆	13.8 ± 3.6	11.3 ± 3.1	10.6 ± 2.3	9.9 ± 0.7	10.6 ± 1.1	10.2 ± 0.4
Dens. estim. (proposed) ⋆	9.6 ± 5.9	6.4 ± 0.7	5.53 ± 0.8	4.5 ± 0.6	3.8 ± 0.3	3.5 ± 0.1

Table 6.2: Mean absolute errors for cell counting in the synthetic cell dataset [90]. *The columns correspond to the different sizes of the training and validation sets. ⋆ indicates the methods that use the same features. The proposed method (bottom line) matches the performance of the previous best method for the same features. The version that does not smooth the difference between the estimated and the GT density when evaluating the solution, performs considerably worse. The method [53] achieves lower counting error in this case, however it uses different and much stronger features learned in a supervised fashion.*

surveillance camera video.

Table 6.2 shows the results on the synthetic cell dataset presented in [90]. The dataset consists of 200 synthetic images of cells in fluorescence microscopy, with an average of 174 $\pm$ 64 cells per image. Between 1 and 32 images are used for training and validation,while testing is performed on a set of 100 images. The procedure is repeated for five different sets of N images, and the mean absolute counting errors and standard deviations are reported for different values of N. We additionally include the result of the method that does not blur the feature channels, and thus uses (6.4) rather than (6.5) as a learning criterion (in the special case of one-hot encoding and $\lambda = 0$, it simply corresponds to averaging the GT density value corresponding to each codeword in the dictionary). The inferior performance of this baseline highlights the importance of the smoothing in (6.5).

It can be seen from Table 6.1 and Table 6.2 that using simple ridge regression never results in a significant drop in performance compared to the more complex approach in [90], and thus, we use it as part of our interactive system (described next) without any compromise on the counting accuracy. Crucially for sustaining interactivity, our simplification yields a dramatic speedup of the learning procedure.

To avoid confusion, we note that ridge regression was proposed as a baseline for counting in [90], as shown in Table 6.2, and tested with the same set of features we have used, but resulting in much poorer performance. This is because, the regression in that baseline was learned at the image level, i.e. it treated each of the training images as a single training example, which resulted in a severe overfitting due to a limited number of examples. This differs from learning w to match the pixel-wise object densities (it can be seen as an extreme case of infinitely wide Gaussian kernel G). Finally, we note a connection between the proposed approach and [53] that also performs smoothing of the output density function (at the testing stage) using structured-output random forests.

6.3 Density estimation on temporal sequences

In Sections 6.1 and 6.2 we discussed the general scenario of counting objects in still images, which is also applicable for time-lapse sequences in a straightforward manner by treating frames independently. Nevertheless, when dealing with time-lapse sequences, it would often be the case (as shown in the experimental Section 6.4), that the intrinsic counting errors of the density estimation appear as "noise" in the curves of object counts over time. Moreover, even if the general trends of the counting curves are maintained, for some application, a high level of noise might result in the loss of relevant information of the population count dynamics such as subtle synchronization of cell cycles in a cell counting experiment, which might not be recovered through post-processing.

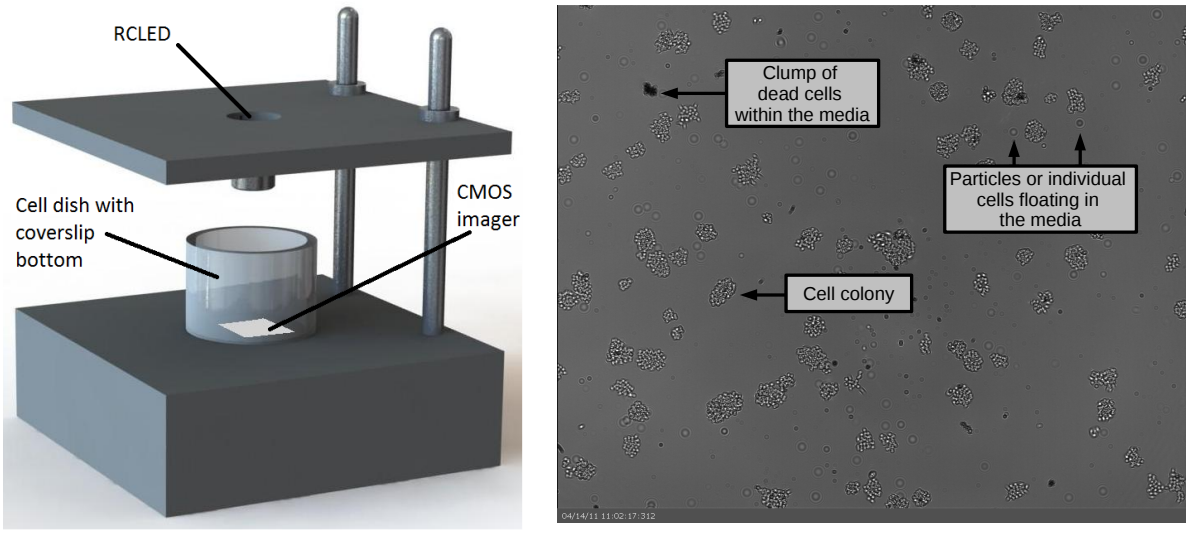
We propose a simple extension for density estimation methods that facilitates learning a mapping that produces counts that are smooth in the temporal dimension, and that is particularly applicable for time-lapse microscopy sequences. The approach is based on

the assumption of low variability between consecutive frames, which suites microscopy applications due to the generally slow changes in time. For example, a time-lapse sequence of an attachable cell culture imaged every 10 min, would contain only minor changes in the local number of cells between two or even three consecutive frames. Therefore, when learning the density estimator w (using the notation presented above), we impose the constraint that the local density estimation must be similar in consecutive frames of the time-lapse sequence. This can be achieved, for example, by setting the first-order or second-order temporal derivatives in the object density to zero between pairs or triplets of consecutive frames, respectively.

The specific way of imposing the temporal smoothness constraint would depend on the learning approach used for the density estimation. In the case of [90], it is possible to add to the training set additional pairs or triplets of consecutive training frames where we establish that $\mathcal{D}_{MESA}(F_t(p), F_{t-1}(p)) = 0$ or $\mathcal{D}_{MESA}(F_{t-1}(p) + F_{t+1}(p), 2 * F_t(p)) = 0$. Similarly, in the case of the density estimation through ridge-regression (Section 6.2), the first- or second-order temporal derivatives can be captured within the observations matrix X in (6.4) by concatenating the direct observations to the difference matrix $X_{first} = |X_t - X_{t-1}|$ or $X_{second} = |-X_{t-1} + 2 * X_t - X_{t+1}|$, and setting the corresponding target values in Y to zero.

Intuitively, the additional training examples containing the temporal difference information will tend to keep those feature channels that vary in consecutive frames; then, their weight in w would decrease during the optimization and those feature channels would have a diminished effect over the density estimation of new examples, resulting in smoothed temporal results. Indeed, the weight of the difference examples w.r.t. the direct examples must be controlled in order to achieve a good balance between smoothness and accuracy, for which we use an additional parameter α. In the experimental section α is set empirically in the validation sets of each of the time-lapse experiments.

Finally, we note that adding training examples for the purpose of temporal smoothing requires no additional annotations as the ground-truth density of the corresponding



(a) *CyMap device* **(b)** *Image example: HeLa cell culture*

Figure 6.1: Lens-free microscope. *For the application of cell counting in lens-free microscopy, we use (a) the CyMap device. The imager (CMOS imaging sensor) is placed in contact with the bottom of the cell dish and illuminated from a point-source LED. The light diffraction patterns created by the sample are then recorded by the imager, where (b) different elements present on the dish can be distinguished. The LED-imager distance is typically ∼44 mm, and the whole device easily fits in a common incubator.*

frames is not used.

6.4 Counting cells in lens-free microscopy

In this experimental section we show the application of the density estimation methods for the particular task of cell culture growth monitoring on large fields of view from lens-free microscopy. Even though the core of the experiments is the counting through density estimation, the overall experimental pipeline requires additional image processing and image analysis modules, such as obtaining the cell count for individual cell colonies throughout the fields of view. Therefore, the different steps of the application pipeline are also described within this section.

This application very much relates to [57], where the MESA-based counting framework [90] was first applied to the same task of cell counting in lens-free microscopy. In comparison to [57], we extend the evaluation on single frames to include the ridge-regression density estimation framework of Section 6.2, automatize several intermediate

steps in the application pipeline such as the dot-annotation of the lens-free images, and show experiments on time-lapse sequences using the framework introduced in Section 6.3.

The lens-free microscopy images and time-lapse sequences where collected in collaboration with Dr. Giselle Flaccavento, Dr. Boris Vojnovic and Dr. James Thompson at the Gray Institute for Cancer Research, University of Oxford, UK.

A detailed overview of the application is presented next.

6.4.1 Application overview

At present, most optical cell imaging is carried out using large and often costly microscopes or high-content automated imaging systems. Such approaches have a great ability to image intracellular features and significant work has been directed towards achieving increased spatial resolution for this purpose. Whilst this is undoubtedly of benefit for the studies of molecular interactions within the cell itself, there are significant research areas where this approach is unnecessary. The requirements in numerous research activities is to track cell positions (e.g. chemotaxis), to monitor cell division (e.g. clonogenic assays) or to monitor movement of confluent groups (e.g. wound healing assays). In order to perform such imaging, a mosaic of many frames is acquired, often at low magnification. The composite image quality is determined by the accuracy of auto-focusing algorithms and the performance of motorized stages as well as by associated software tools to "stitch" the individual images. Such relatively complex approaches require the microscope to be fitted with large incubation housings which ensure appropriate cell growth conditions and microscope system stability.

Recently, lens-free diffraction imaging devices have been used to acquire images of cells sparsely seeded on glass slides [73] and confluent cells [174]. Due to their compactness, these kinds of microscopy devices considerably simplify the arrangements required to capture cell images in the appropriate cell growth conditions. In this application, we make use of the large field of view and lens-free device called CyMap [160], a compact microscope able to acquire time-lapse images from within an incubator. The CyMap

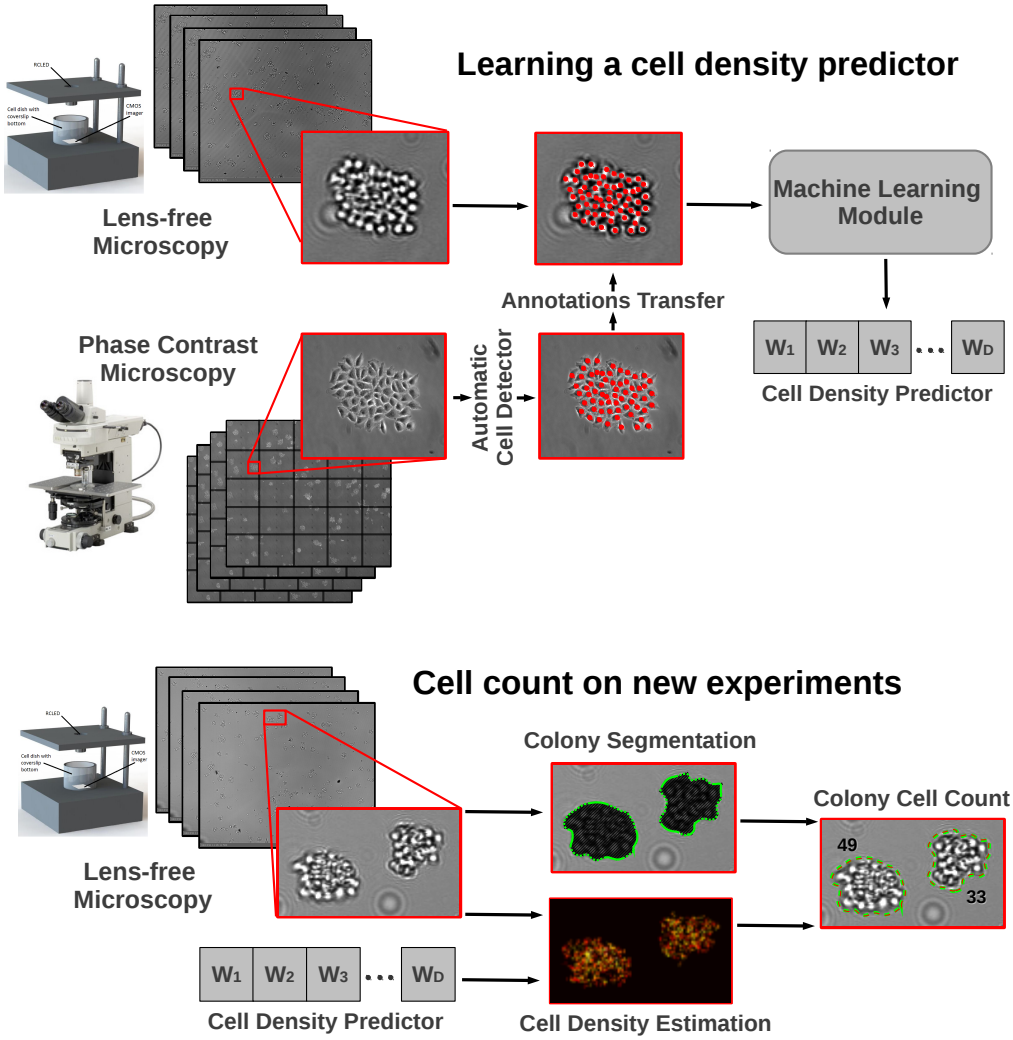


Figure 6.2: *General training (top) and testing (bottom) procedures for the lens-free microscopy application. The lens-free images from the CyMap are annotated by transferring dot-annotations from phase contrast images of the same dishes, where cells are clearly discernible due to the much higher magnification. Using the dot-annotations, a density estimator (W) is computed, which is then used to count the number of cells in the (segmented) colonies of unseen images. See text for details on the different steps of the process.*

device, shown in Figure 6.1a, operates by illuminating the sample with a single LED, and capturing the light diffraction patterns on a CMOS sensor located below the sample, resulting in the type of images shown in Figure 6.1b. In this case, not only individual cells might no be visible, but this also fits the scenario discussed in Chapter 2 where simple count estimation methods such as area-based or template matching fail to obtain an accurate cell counts throughout the time-lapse sequence. Therefore, we adapt the density estimation methods discussed in this chapter in order to monitor the growth of the individual HeLa cell colonies, which can allow us to quantify, for example, the growth rate of the colonies as a response to external stimuli (e.g. radiation therapy) similar to clonogenic assays.

Aside from the density estimation learning and testing stages, this application requires additional steps. First, the application requires to obtain the image dot-annotations, but this cannot be done directly on the large field of view lens-free microscopy image as individual cells are no visible in this modality for relatively dense dishes. Therefore, a second microscopy modality is used to produce the annotations of the training dishes, which are then transferred into the lens-free microscopy images through image registration. Secondly, it requires the correction of the illumination artifacts of the lens-free microscopy images. Finally, as the growth monitoring needs to be done on a per colony basis, it is required to segment individual cell colonies. The general training and testing procedures are illustrated in Figure 6.2, and all of the modules are described next.

6.4.2 CyMap image normalization

Although the illumination in the CyMap is reasonably even, the construction is such that the illuminating LED and its simple beam angle limiting aperture are not necessarily coaxial with the centre of the imager. This causes intensity variations across the image and we thus normalize the image to create an even illumination using a three step process. Firstly, the cell colonies are segmented in the CyMap image, (see Section 6.4.4 for details), leaving only a cell-free background. Next, the portions of the CyMap image that contain

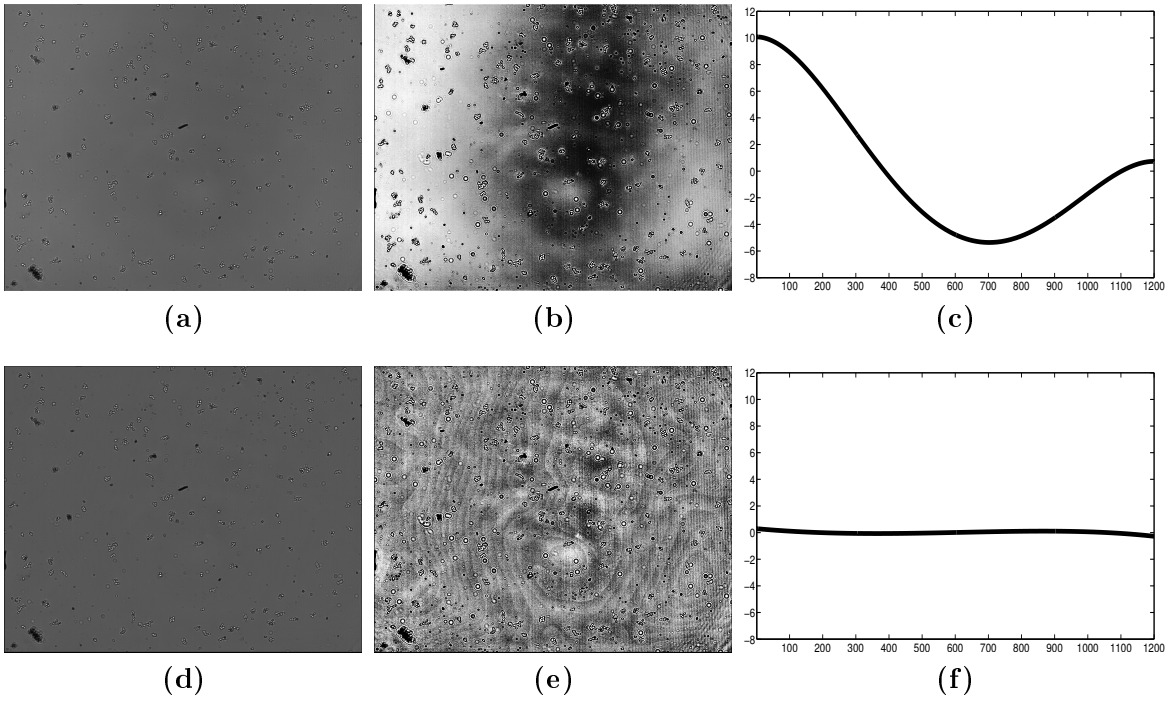


Figure 6.3: *The original CyMap image (a) has uneven illumination caused by the LED during CyMap image acquisition, as can be seen clearly if the histogram of the image is equalized (b), as well as in the intensity profile of a cross-section of the background (c). After normalization (d), the illumination is even throughout the image (e,f).*

the colonies are estimated using natural neighbour interpolation and a pseudo background is created. Finally, Gaussian filtering is applied to ensure that a smooth transition is achieved between the actual and pseudo background portions of the images. The original CyMap image is then pixel-wise divided by this pseudo background image to produce a normalized image. Figure 6.3 shows an example of an acquired CyMap image, before and after normalization.

6.4.3 Image encoding

Each pixel p in each training image $i \in \mathcal{I}$ is encoded with a real-valued feature vector x_p^i from a high-dimensional feature space R^N. The encoding should contain enough information about the local patterns so that distinguishing between image parts with different cell density is possible. To achieve this, we use two types of features, which we refer to as *local* and *contextual* features respectively. The local features try to capture the fine texture in the near surroundings of the pixels, as this varies with cell density in the CyMap application. The contextual features capture coarser information related to the location of a pixel within a cell colony.

The local features correspond to a texture representation borrowed from [157], which consists of the raw intensity values of the pixels in the 9 x 9 pixels patch centred on the pixel p, which is then contrast-normalized over this neighbourhood. This results in an 81-dimensional descriptor vector for each pixel in the image. The contextual features correspond to the response to a filter bank consisting of Gaussian kernels, difference of Gaussians and Gabor filters; 20 different filters are applied separately to the entire image, and the response to each filter in each pixel is concatenated, resulting in a 20-dimensional descriptor vector per pixel. The two feature vectors (local and contextual) are then quantized individually to obtain a sparse representation using the unsupervised clustering algorithm (k-means), with 2048 means for each of the two vectors. The result of the quantization is two sparse 128-dimensional vectors (each of the vectors contain 127 zeros and 1 “one" corresponding to the number of the closest prototype). The two vectors

are concatenated to obtain a final 256-dimensional representation for each pixel.

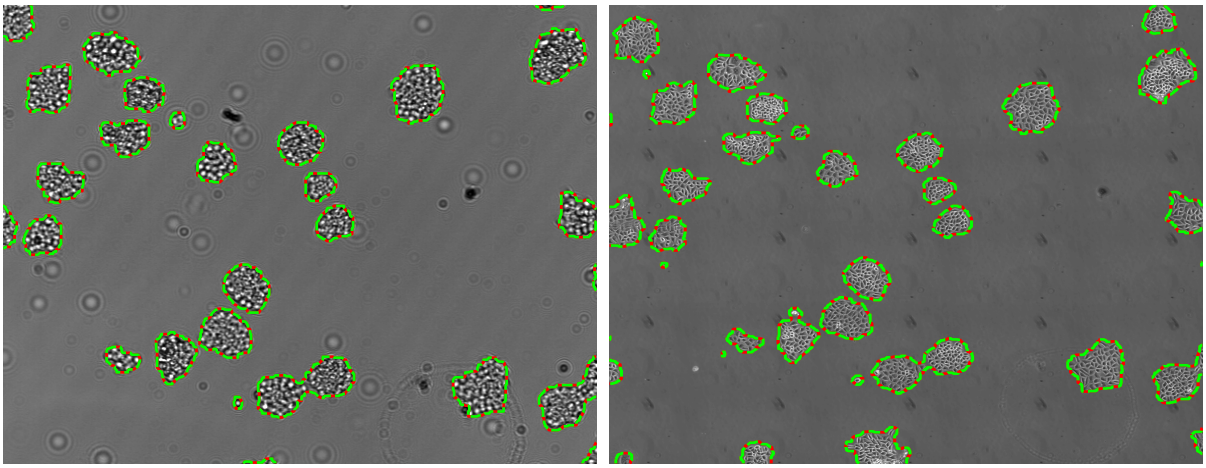
6.4.4 Image annotation

For the purposes of supervised training of the machine learning algorithm and quantitative evaluation of the results, annotations showing the location of each cell within the images are required. However, as can be noted in Figure 6.1b, annotating CyMap images directly is not always possible, especially as the cell density in the colonies increases. Therefore, at the system training stage, the support of a secondary microscopy modality is necessary, for which phase contrast microscopy was chosen. Specifically, a mosaic of 90 phase contrast images, acquired using a 10x objective, is required to cover approximately the same field of view as a CyMap image; we refer to the mosaic as a phase contrast frame.

Although the annotations required are simple (approximate cell centre locations), and the procedure needs to be done only once (the learned vector will work for any data collected under the same experimental setup), the amount of data that needs to be annotated is large, thus we also automate the phase contrast annotation process. Assuming that we have correspondent CyMap and phase contrast frames, we automatically annotate the CyMap image with the following procedure:

- Identify and mark cells in the phase contrast composite image to create annotations indicating the coordinates of each cell.
- Colony Level Segmentation and centroid determination in both the phase contrast composite image and the CyMap image.
- Image Alignment by computing the transformation between the phase contrast composite image and the CyMap image using the colony centroids as corresponding points.

Once these steps are performed, transfer of the annotations from the phase contrast composite image to the CyMap image can be performed using the computed alignment transformation. Each step of the automatic annotation procedure is now briefly described.



(a) (b)

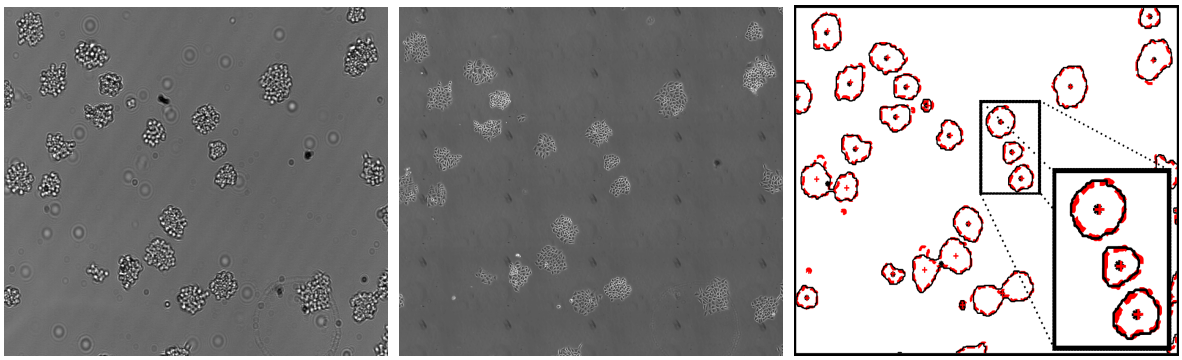
Figure 6.4: *Example results of the colony segmentation in (a) CyMap and (b) phase contrast images, indicated with dashed lines.*

Automatic annotations in phase contrast images

To automatically annotate the phase contrast frames containing HeLa cell colonies, we use the cell detection method presented in Chapter 3. In fact, the phase contrast dataset presented in Section 3.1 (Figure 3.1c) correspond to a subset of the phase contrast images collected for the CyMap colony growth monitoring application.

Colony-level segmentation

In the annotation procedure, the goal of the colony level segmentation is to calculate the centroids of each colony within the phase contrast and CyMap frames in order to create a registration between the two equivalent views. The strategy followed to segment CyMap images on a colony level was presented in [57], but is also applicable to phase contrast images. The method consists of training a random forest classifier [26] to differentiate between pixels that belong to a colony and pixels that are part of the background of the image, whereas the output of the classifier is a class probability. This probability is used as an input to an optimization framework based on graph-cuts [24] that computes a piecewise smooth segmentation. Example colony segmentation results for both microscopy modalities are shown in Figure 6.4.



(a) (b) (c)

Figure 6.5: *An example of the colony registration results: (a) fragment of the CyMap image, (b) fragment of the phase contrast image, and (c) overlay of the boundaries of the colonies after the CyMap and phase contrast images have been aligned. The black solid lines and red dashed lines correspond to the boundaries of the colonies in the CyMap and phase contrast images respectively, and their centroids are indicated, in the same order, with stars and crosses. Due to the robust estimation of the parameters in the affine transformation, the alignment is accurate even in the presence of segmentation errors.*

Image alignment

The mapping of the colonies in the phase contrast frame to those in the CyMap frame is treated as an alignment problem where the correspondence between the colonies in each image modality needs to be found. An affine transformation was chosen to account for translation, rotation and scaling [67]; the scaling being non-isotropic to cope with perturbation due to differences in projection between modalities.

Errors in the colony detection, possibly caused by fibers or dead cells within the cell medium, can result in erroneous reference points; a robust estimation algorithm is thus required to fit the image transformation. The Random Sample Consensus algorithm (*RANSAC*) [54] is employed for this task. A typical result of the alignment between a lens-free image and a phase contrast frame is shown in Figure 6.5. Once the mapping between frames is obtained, any annotation on the phase contrast frame can be accurately transferred to the CyMap frame.

6.4.5 Experimental Validation

In order to validate the usage of density estimation methods for this application, a data set of cell cultures at different stages of growth was collected with the CyMap device. This data was used to test the counting accuracy of the learning-based method, and to compare it with classical approaches for counting cells in colonies.

A total of 20 cell dishes of HeLa cells were seeded, each incubated for 2 to 7 days before imaging with the CyMap. The number of cells within the field of view varied between 600 and 3500 for the 20 cell dishes. The number of colonies within the imager field of view varied between 64 and 118 with the largest single colonies containing approximately 70 cells. Where two or more colonies had merged into one, there were as many as 250 cells per colony.

The 20 dishes were annotated both manually and automatically (with the procedure described in Section 6.4.4 using the phase contrast mosaics), and used for training and testing with a 4-fold cross-validation procedure. That is, the data set was split in four groups (5 dishes per group); one of the groups was retained for testing, while the other 3 were used to learn the density estimation vector w, using the automatic annotations or manual annotations as ground truth. The testing procedure was then applied to the group left out, and the counting results were compared to the manual annotations. This process was repeated for each of the four groups, allowing the evaluation of the counting performance in the entire data set.

The results of three alternative cell counting methods were compared with the counting approach of density estimation using both, the MESA distance [90] and the ridge regression alternative from Section 6.2. For each of the methods, the segmentation of the colony, was used to delimit the boundaries of the colonies. This identical segmentation was used for each of the methods to allow a comparable evaluation between techniques. Each of these baseline methods require the optimization of one or more parameters, for which the 4-fold cross validation procedure described for the learning method was also

used.

The mean absolute counting error was calculated for each of the methods on a per image basis, by taking the mean of the absolute difference between the ground truth cell count and the estimated cell count in each of the colonies detected by the colony segmentation algorithm.

Template matching method

A Laplacian of Gaussian (LoG) filter, with σ and a filter radius defined through cross-validation, was applied to the normalized CyMap images. A peaked response was produced, with high intensity values indicating locations where the LoG filter was matched with the shape of cells in the CyMap image. Next, these peaks were detected with an intensity threshold and the number of pixel clusters appearing in each of the colony segmentations was determined and counted as individual cells. For colony sections of higher cell density, the size and appearance of cells tend to change and thus the template matching algorithm would increase its error rate, mainly through undercounting.

Intensity threshold method

An intensity threshold was applied to each of the segmented CyMap images with the goal of finding locations of cells, which generally appear as bright spots in the images. After thresholding, the centroid of each high intensity region of connected pixels was identified, indicating the location of cells with high intensity responses. Cells omitted from the cell detection occurred when the responses of cells produced a low intensity level in the CyMap image or when two cells were very near each other. In both of these cases, the centroids of the bright regions in the CyMap image did not represent the true number of cells, creating errors in the cell counts. The intensity threshold method depends strongly on the intensities in the CyMap images after normalization. For cell dishes that have larger colonies, there is more variation in the colony sizes and cell sizes. These variations produce CyMap images with varying peak intensities at the cell locations. For this reason,

using one intensity threshold for a set of CyMap images did not produce optimal results. Additionally, when cells were very close together, as in larger colonies, a high intensity area was produced covering the pixel locations of multiple cells. Multiple cells were counted as one cell in these cases due to the binary nature of the threshold results.

Area-based method

An area-based cell counting method was applied to the CyMap validation images to estimate cell numbers in each colony without identifying individual cells. A second-degree polynomial function was fitted between the size of imaged colonies and the ground truth cell counts using least squares minimization and the 4-fold cross validation procedure. For each CyMap image, the area-based cell counting was calculated using the relationship $Count_{area-based} = P_2 \cdot Area^2_{colony} + P_1 \cdot Area_{colony} + P_0$, where the coefficients P correspond to the best fit curves in the colony area [pixels^2] versus ground truth cell count relation.

The area-based method is not able to accurately count cells in colonies with varying cell sizes. As shown in Figure 6.8, within one CyMap image, or even within one colony, the cell 'areas' may vary significantly, particularly in larger colonies which often contain smaller cells in the interior of the colony and larger cells along the colony perimeter.

6.4.6 Results on static frames

An example of a CyMap image after being processed with a density-based cell counting method is shown in Figure 6.6.

The counting error for the five cell counting methods; the template matching, the threshold method, the area-based method, and the two density estimation methods, each with manual and automatic annotations, are shown in Figure 6.7 for each of the 20 CyMap images. The results have been arranged in increasing order with respect to the mean colony size in each image, highlighting the fact that cell counting error increases with colony size for all of the methods.

A scatter plot of the ground truth cell count versus each cell counting method count

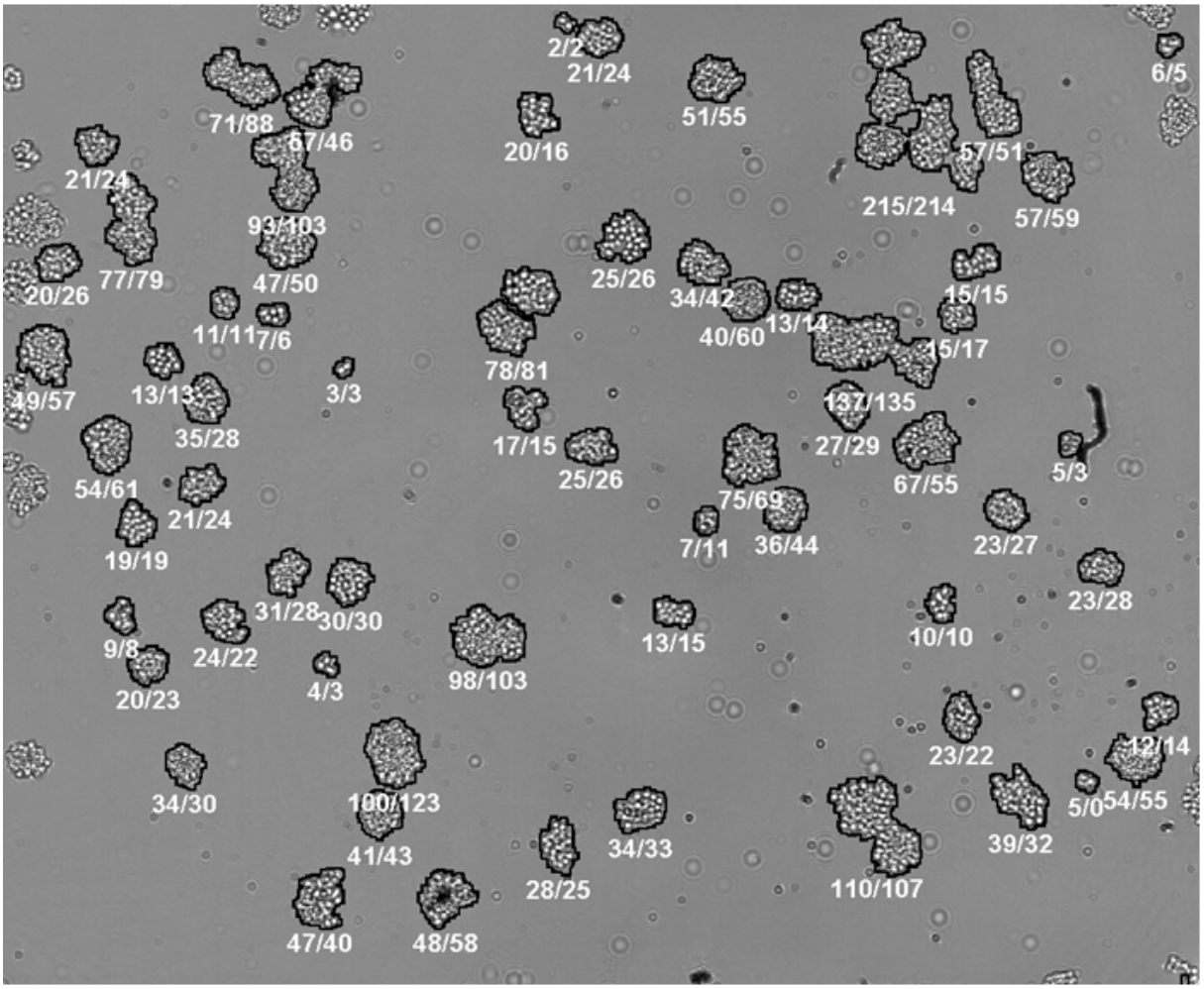


Figure 6.6: *Example CyMap cell counting results using density estimation through MESA distance. The average colony size for this image is 40 cells. The density estimation cell counts and the ground truth cell counts for the colonies within these images are shown below each colony. Partial colonies at image edges are discarded.*

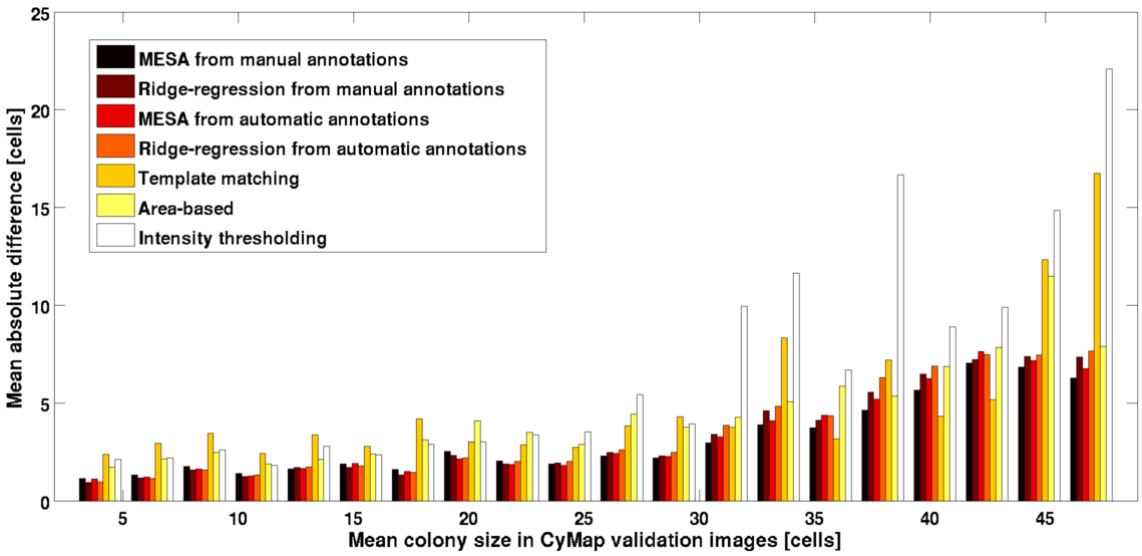


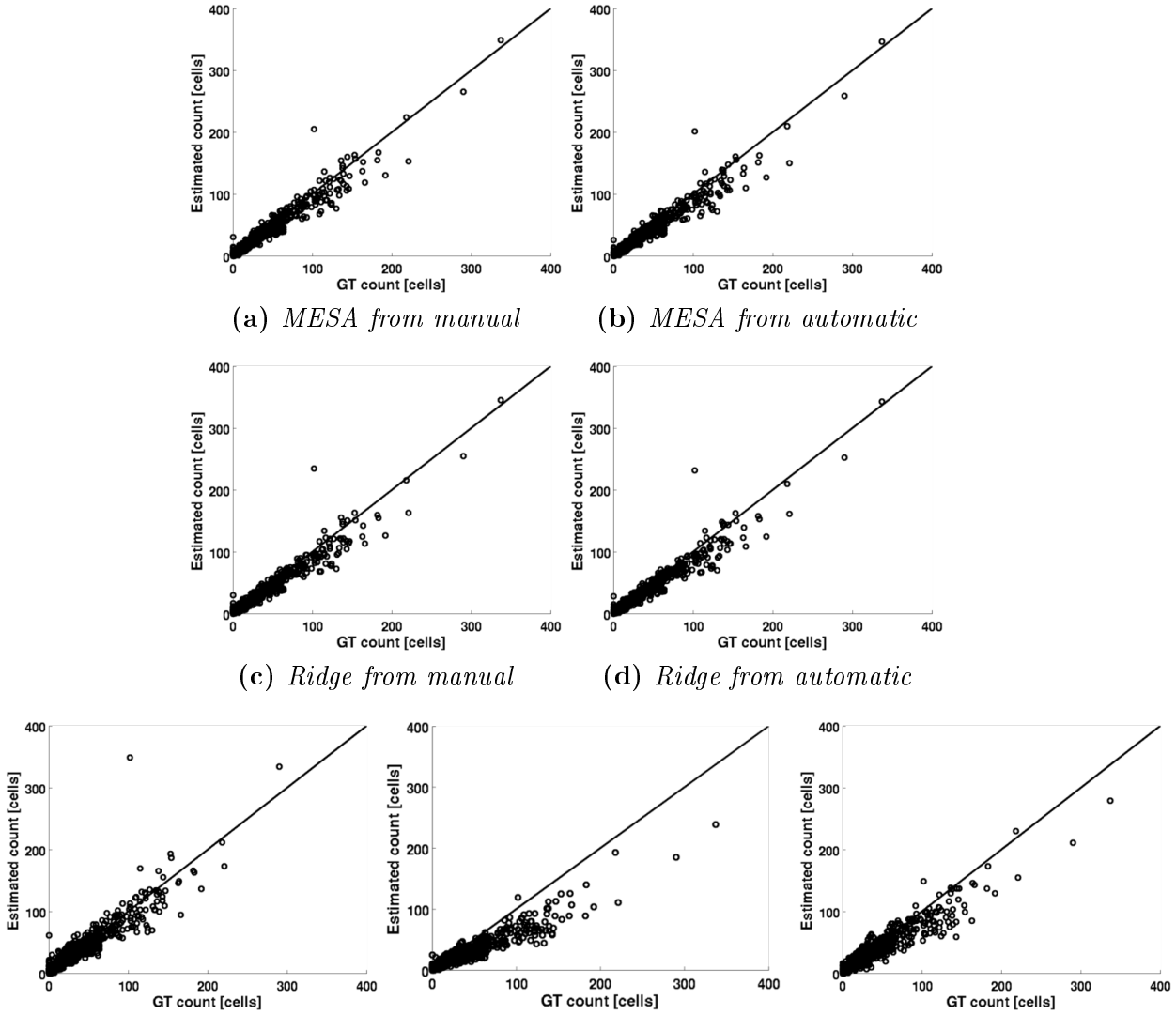
Figure 6.7: *Results of the five cell counting methods for each cell dish. The machine learning cell counting method produces the lowest errors for every cell dish. The cell counting errors for the template matching, area-based and threshold methods are particularly large for cell dishes containing large colonies. No significant differences can be seen between the different density estimation methods, or between the cases where the learning is done from manual or automatic annotations.*

is shown in Figure 6.8. A line showing perfect agreement between the cell counting method and the ground truth cell counts is shown for each of the plots. As the colony size increases, so does the deviation from this line.

6.4.7 Experiment on time-lapse sequences of lens-free microscopy

We now extend the application of density estimation for the use-case of counting cells in time-lapse sequences of lens-free microscopy, in which we also show the use of the method to smooth temporal density estimation presented in Section 6.3.

For this experiment, we use time-lapse sequences collected under the same experimental setup described in Section 6.4.1, and the main goal is to assess whether cell counting through density estimation can be used to perform clonogenic assays for radiation experiments. The traditional clonogenic assay [58] is an experimental technique used to quantify the effect of an external stimuli, such as a drug or radiation, over the proliferation of cells, and it is widely used in cancer treatment research. Nevertheless, it is a very time-consuming technique due to the cell incubation time required to assess



(a) *MESA from manual* (b) *MESA from automatic*

(c) *Ridge from manual* (d) *Ridge from automatic*

(e) *Colony area* (f) *Thresholding* (g) *Template matching*

Figure 6.8: *Scatter plots of the the cell counting method counts versus the ground truth cell count. A line showing perfect agreement between the cell counting method and the ground truth cell counts is shown for each of the plots; the deviation from this line shows the error in counting. The two variations of the annotation method for each of the two density estimation methods are shown, namely MESA distance from (a) manually and (b) automatically annotated training images, repeated for the ridge regression method (c,d). Density estimation methods produce results that better agree with the true cell count in the entire range of colony sizes, when compare to the base-line counting methods based on (e) colony area, (f) thresholding and (g) template matching.*

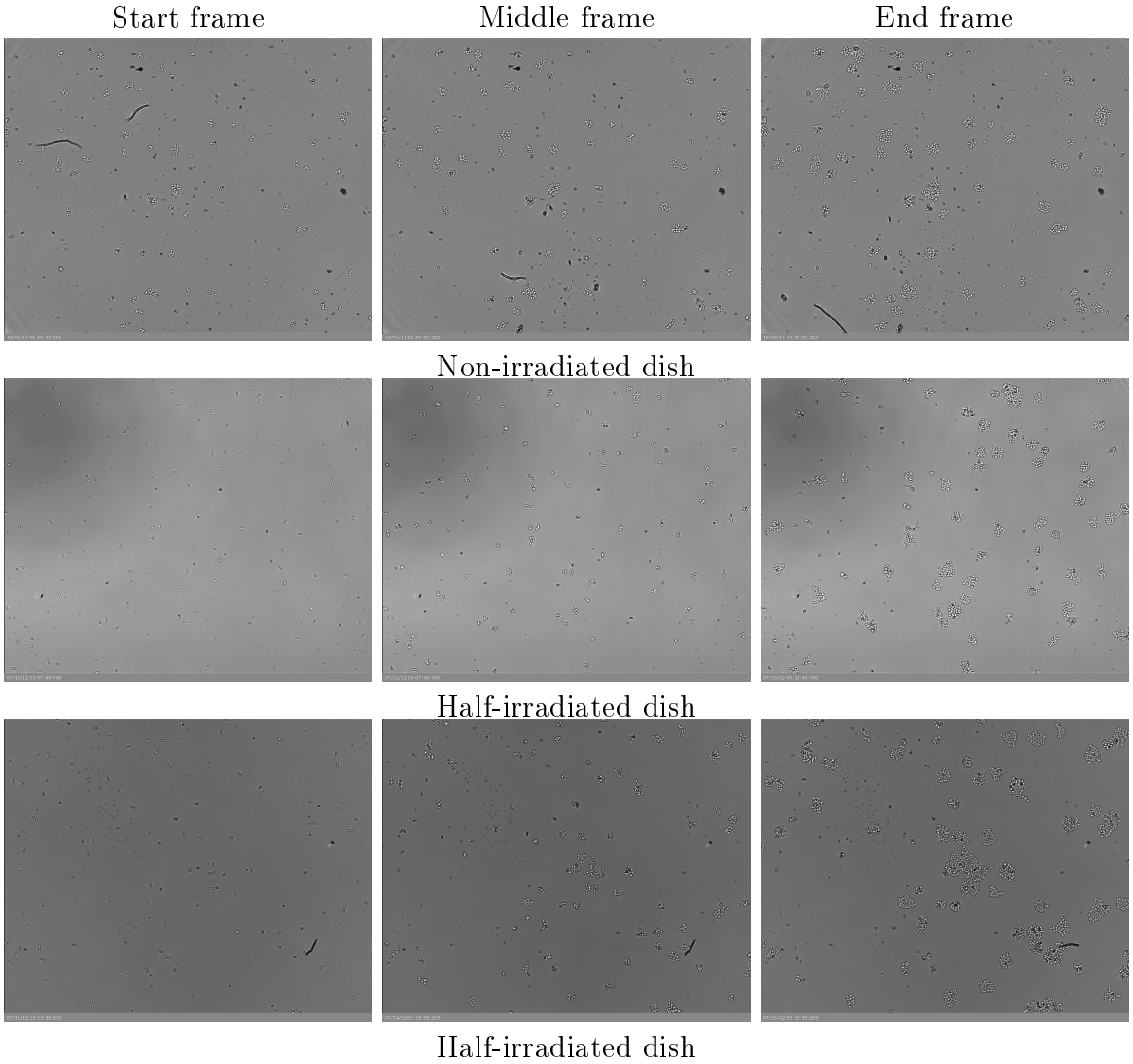


Figure 6.9: *Example frames of the three time-lapse sequences of lens-free microscopy, where the initial (*left*), middle (*middle*) and final (*right*) frames are shown.*

the survivability of the cell cultures, as well as the high number of repetitions required to achieve statistically significant results. For this type of assay, time-lapse information has the potential of improving the experimental throughput by recognizing earlier the survivability of cell colonies. Furthermore, doing so over large fields of view improves the statistical robustness of the experiment, and compact lens-free devices enable the possibility of running multiple experiments in parallel from within an incubator.

Within this experimental section, however, we only show a proof of concept with qualitative results due to the current unavailability of the required data to show the entire clonogenic assay application, and the lack of ground truth counts for the frames in the temporal sequences. Instead, we focus on the general applicability of density estimation methods to monitor cell colony growth in time.

Three time-lapse sequences are used, which consist of a non-irradiated cell dish, and two dishes half-irradiated with a dose of 2.8Gy. The dishes were imaged every 10 minutes, for an approximate of 5 days. Example images are shown in Figure 6.9 from the initial, middle and final state of each of the three dishes.

For each of the three sequences, we show through Figures 6.10-6.12 the cell count on a selection of colonies over time. Specifically, we show the colony growth curves using six different density estimation variants: MESA-distance [90], ridge-regression (Section 6.2), and both density estimation methods with first- and second-order temporal smoothing as described in Section 6.3.

In order to produce the colony count in time, it is necessary to track each of the colonies throughout the temporal sequence. Nevertheless, as the HeLa cell line remains attached to the dish and not moving, tracking colonies is straightforward and it is done through nearest neighbour matching.

Figure 6.10 shows the results of counting on the non-irradiated dish over the time-lapse sequence. The colony growth curves show that it is possible to track the cell proliferation per colony, and that using the density estimation methods with the smoothness constraint, can significantly reduce the count variation from frame to frame. However, no significant

difference can be observed between the two variants of the smoothness constraint, namely, the first- and second-order temporal differences.

Figures 6.11 and 6.12 show the case of counting in the half-irradiated dishes, where the plotted curves correspond to colonies in the irradiated and non-irradiated side of the dish, color-coded with red and blue respectively. Despite the potential experimental errors related to precisely irradiating half of the dish[1], the growth curves show a clear differentiation in the proliferation rate between the two different populations. Moreover, such differentiation can be perceived relatively early in the incubation period; approximately 4 days in the examples shown. Although the presented evidence is limited, the indication that colonies do not need to be fully grown (i.e. 1 to 3 weeks [58]) to assess the survivability could have an impact in the throughput of clonogenic assays.

6.4.8 Discussion of lens-free microscopy experiments

The nature of the CyMap images presents many cell counting challenges; mainly, the variation of cell size and appearance within colonies. Large cells, typically included in a colony with a low cell density and therefore low cell per pixel density, were accurately counted using all cell counting methods. However, as the cell density increases, so does the overlap between diffraction patterns from each cell and the difficulty of the counting task. Using the template matching and intensity thresholding cell counting methods, a large increase in errors was observed for colonies with overlapping diffraction patterns. The colony area, although generally increasing as the cell culture grows, is not an accurate indication of cell count either, as relation between the colony size and the cell count is not only non-linear, but also presents an increasing variance as the cell density increases. In contrast, during training of the density estimation methods, the intrinsic properties of the CyMap images are taken into consideration, making this approach more robust to such image aberrations and thus suitable for all stages of colony growth.

The baseline methods are also highly dependent on the pre-processing steps. For

[1]The experimental setup of irradiating half of the dish was created for illustration purposes. In practice, different dishes are fully radiated at different doses, and the cell survivability

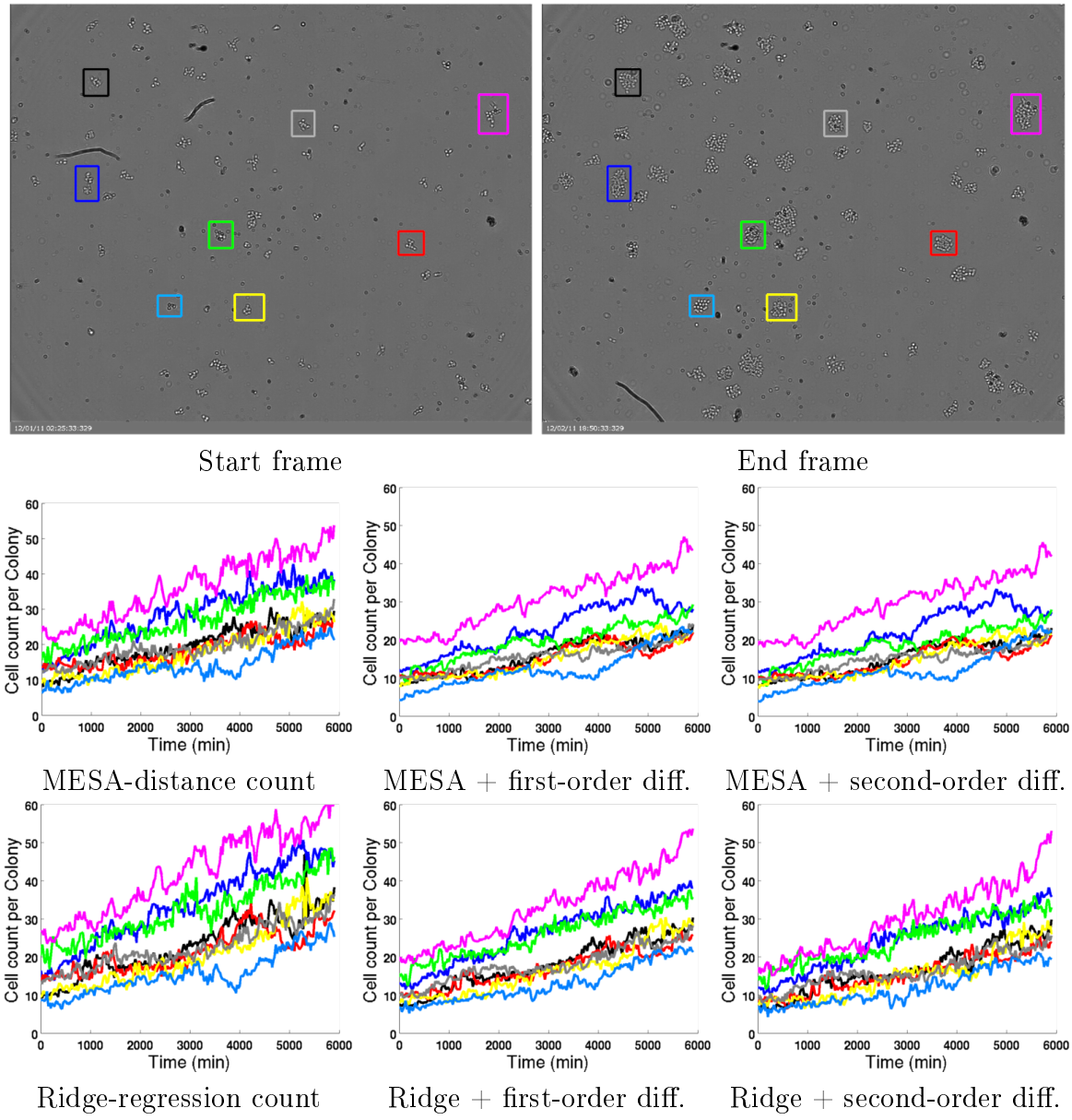


Figure 6.10: *Counting on the non-irradiated temporal sequence. The top row shows the initial and final state of the time-lapse sequence, with bounding boxes showing the color-coded colonies selected for plotting. In both frames, we show the bounding box of the colony extracted from the final frame in order to clarify the correspondence between colonies in both frames. In the middle and bottom rows, we show plots of cell count vs. time using six different variants of the density estimation-based counting: MESA-distance [90], ridge-regression (Section 6.2), and both density estimation methods with smoothing based on the first- and second-order temporal differences as described in Section 6.3.*

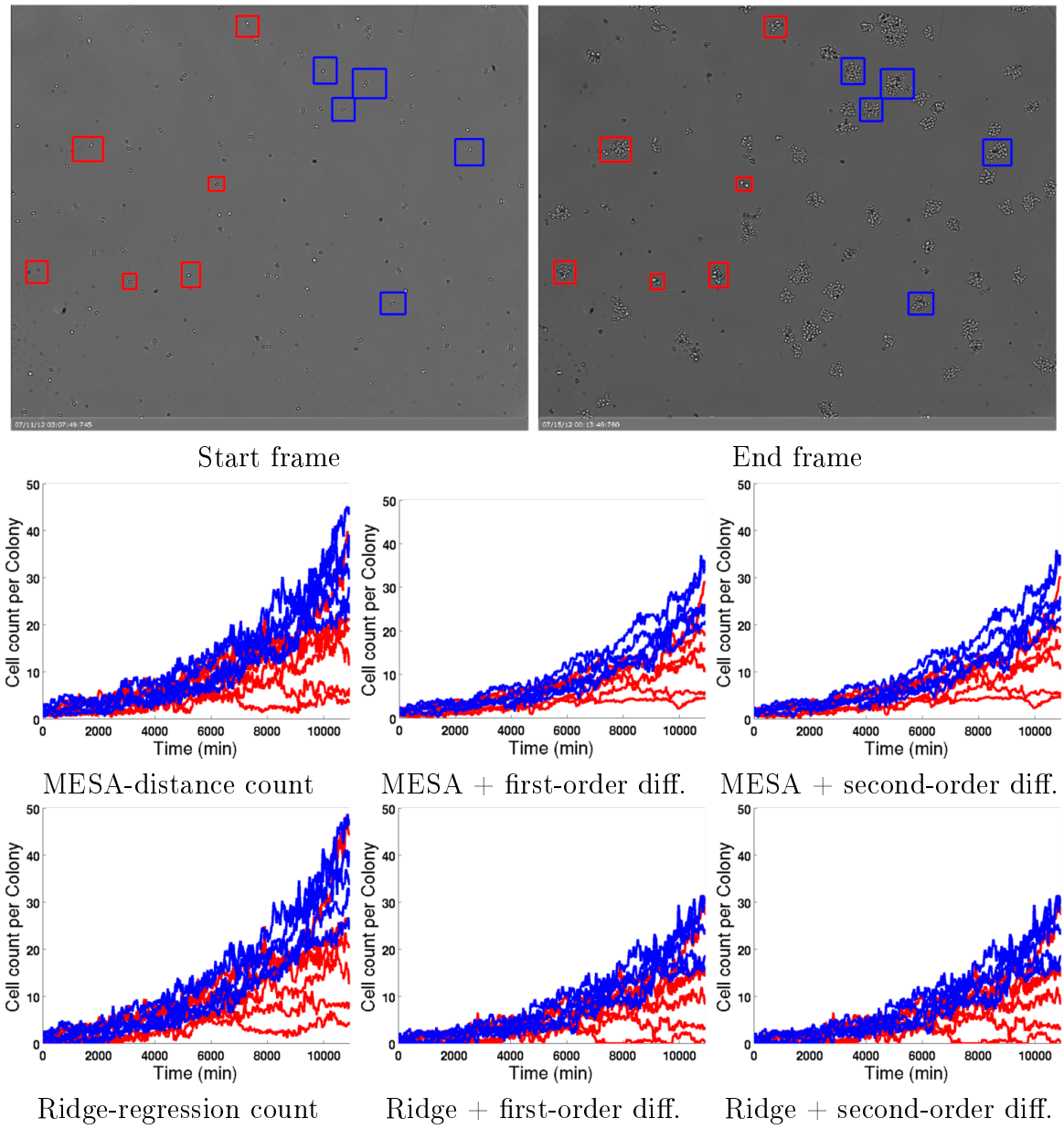


Figure 6.11: *Counting on a half irradiated temporal sequence. The top row shows the initial and final state of the time-lapse sequence, with bounding boxes showing the colonies selected for plotting. The colonies with the blue bounding boxes belong to the non-irradiated side of the dish* (right)*, whereas the colonies with the red bounding boxes belong to the side irradiated at 2.8Gy* (left)*. In both frames, we show the bounding box of the colony extracted from the final frame in order to clarify the correspondence between colonies in both frames. In the middle and bottom rows, we show plots of cell count vs. time using six different variants of the density estimation-based counting: MESA-distance [90], ridge-regression (Section 6.2), and both density estimation methods with smoothing based on the first- and second-order temporal differences as described in Section 6.3.*

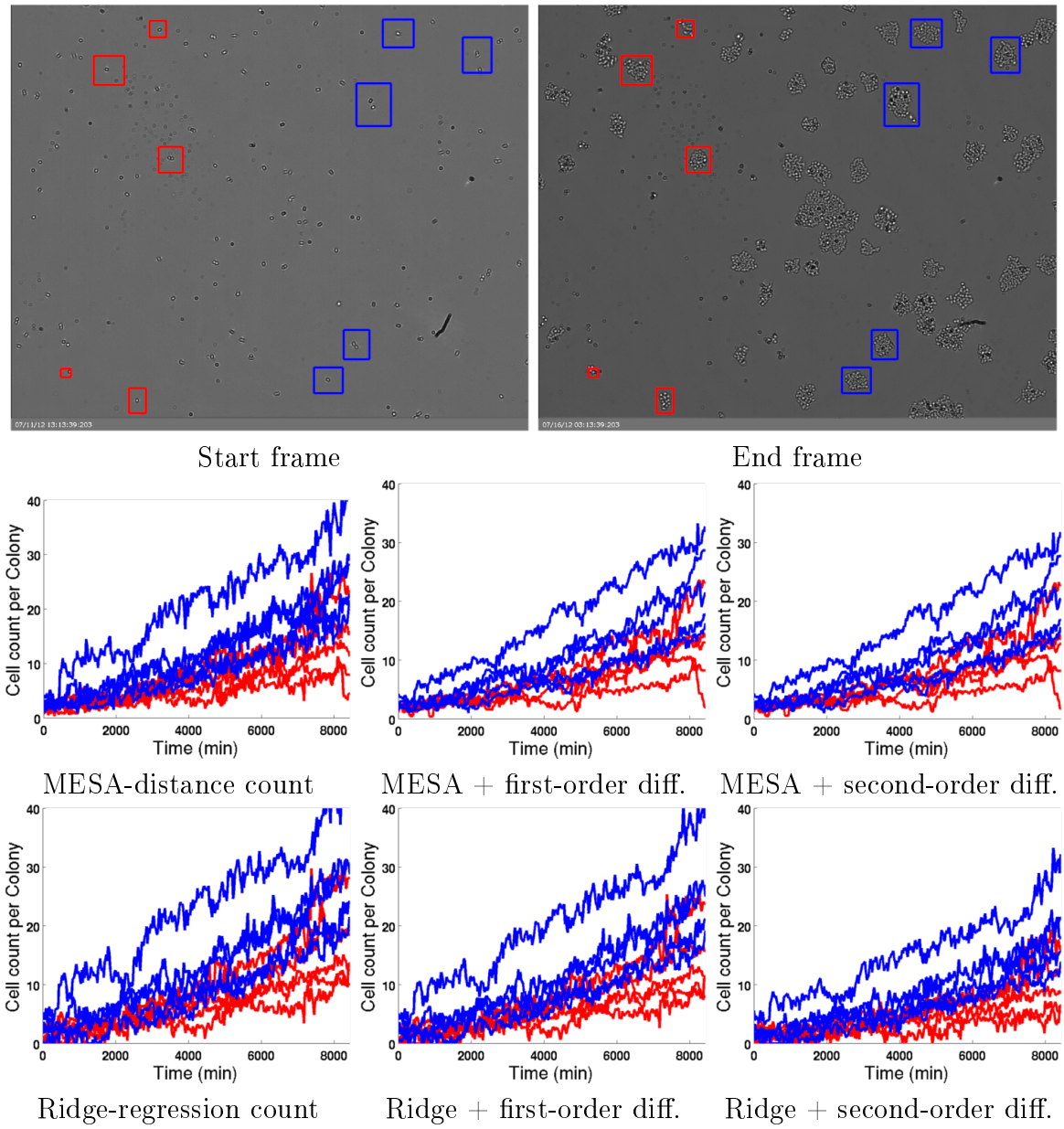


Figure 6.12: *Second example of counting on a half irradiated temporal sequence. The top row shows the initial and final state of the time-lapse sequence, with bounding boxes showing the colonies selected for plotting. The colonies with the blue bounding boxes belong to the non-irradiated side of the dish (*right*), whereas the colonies with the red bounding boxes belong to the side irradiated at 2.8Gy (*left*). In both frames, we show the bounding box of the colony extracted from the final frame in order to clarify the correspondence between colonies in both frames. In the middle and bottom rows, we show plots of cell count vs. time using six different variants of the density estimation-based counting: MESA-distance [90], ridge-regression (Section 6.2), and both density estimation methods with smoothing based on the first- and second-order temporal differences as described in Section 6.3.*

example, the template matching and intensity threshold counting methods are highly dependent on the normalization of the CyMap as they rely on intensity information. Furthermore, all the baseline methods depend on the accuracy of the segmentation algorithm to produce reasonable results as they cannot discriminate between cells and similar objects in the image. Diffraction patterns of particles in the cell media, dead cells, and substrate texture, all produce contributions in the CyMap images. Although changing the medium periodically and filtering it could remove many of these extra particles, such interruptions could also affect cell growth or the completeness of any time-lapse data series. Because the learning-based methods rely on the training provided by the annotated images indicating cells only, discriminative capabilities are intrinsic and these additional particles often result in very low cell density areas with a small effect over the global count.

Regarding the practicality of the methods, it was observed that, although manually annotating the training images for the learning-based algorithm resulted in the lowest counting error obtained, such annotation process can be very time consuming and tedious. In contrast, the ease of using a simple cell detector in conjunction with a colony segmentation algorithm for image annotations resulted in comparably accurate result, but with highly reduced human effort.

Finally, it was observed that the learned density estimators could be directly applied into temporal sequences but this resulted in noisy count estimates in time. Nevertheless, this problem was improved by learning density estimators constraint to temporal smoothness without the necessity of post-processing the temporal counts, which could still be done if required. In the time-lapse sequences form the CyMap lens-free device, no significant difference was observed between the two different ways used to impose the temporal smoothness constraint on either the MESA-distance or ridge-regression counting methods.

6.5 Summary and limitations

In this chapter, we explored the counting of objects in microscopy images through object density estimation, inspired by the method of Lempitsky and Zisserman [90]. We proposed a simpler alternative based on ridge regression and showed that it maintains the performance of [90], while being much faster to compute due to its closed-form solution. Additionally, we explored the case of using the density estimation methods to perform counting over time, for which we proposed a simple and general method for imposing different degrees of temporal smoothness.

We then presented an extensive application of density estimation methods and temporal count for a lens-free microscopy system (CyMap), where indirect object counting is a necessity. This unique combination of a simple imaging device and sophisticated counting algorithm allows robust colony cell counts to be made from inside an incubator at regular and frequent intervals compared to traditional microscopy. In that situation, cells can grow more naturally and can be monitored remotely, without the perturbation of a manual monitoring procedure.

The system can provide researchers with a tool that learns from the user how to asses a specific experiment (training only once), and perform accurately and automatically in consecutive experiments with the same experimental setup, for instance, the same cell line, dish type and cell media. The fact that the counting method in the system is based on machine learning, reduces the input required from the researcher to a task related to his/her area of expertise; in this case, counting cells. This contrasts to using traditional image analysis techniques, where the user needs to tune parameters that are often complex, especially for the non-expert in image analysis (i.e. filter parameters or weighting coefficients), and generally needs regular intervention from the user due to poor generalization.

As shown in Section 6.4.7, the entire microscopy system (i.e. hardware and counting method) can be applied to experiments that require monitoring cell culture growth over

time; for instance, to assess the response of a population of cancer cells treated with radiation or drugs. Moreover, due to the cost effectiveness and compactness of the device, multiple experiments could be performed and analysed in parallel from within the same incubator. Besides the laboratoy settings, this technology and software combination has applications in industrial cell growth, as well as in field situations where cumbersome microscopy hardware cannot be used.

One possible extension of the counting methods discussed in this chapter is the lack of a measure of uncertainty, as in the random forest-based density estimation method of Fiaschi *et al* [53], which would add statistical robustness to an application such as the clonogenic assay. However, in the case of the ridge-regression, a possible an straightforward measure of uncertainty can be obtained from its interpretation as a Gaussian process regression [123].

Finally, for cases where it is still possible to dot-annotate the images directly for counting (i.e. the individual instances of the object are still visible), the question of how much annotation is required for certain application is key for a potential end-user. A possible answer is an interactive system that can give an indication of the accuracy of the results as the user annotates the image. We explore such interactive system next, in Chapter 7.

Chapter 7: Interactive object counting

In Chapter 6 it was shown that object density maps can be estimated simply, accurately and efficiently using ridge regression, and this matched the counting accuracy of the much more costly learning-to-count method of Lempitsky and Zisserman [90]. Taking advantage of the speed at which the regression required for the density estimation can be computed, we now present an interactive counting system, along with solutions for its remaining components.

Generally, an interactive counting system works in the following way; first, the user annotates (e.g. with dots) the objects of interest, but only within a representative but potentially small part of an image and then the system propagates the annotations to the rest of the image. The counting results across the image are then presented to the user, who then has the option to annotate another part of the image where the system has made significant errors. When the user is satisfied with the result, the system provides the count of the objects in the image. A formal overview of the interactive counting framework is presented in Section 7.1.

One key aspect of density-based interactive counting is that density *per se* is not informative for a human user and cannot be used directly to verify the accuracy of the counting (and to provide feedback). Therefore, we propose in Section 7.3 two ways to visualize the estimated density, so that counting mistakes are easily identifiable by the user. This allows our system to incorporate user feedback in an interactive regime according to the user's own criterion of goodness. A second key aspect is that it is not possible to know in advance how many annotations the user will provide, thus risking severe under- or over-fitting if the size of the visual descriptors is pre-defined. For this,

we propose in Section 7.2 an online codebook learning that, in real-time, re-estimates low-level feature encoding as the user annotates progressively larger parts of an image.

We note that even though we mostly deal with dot-annotated isotropic objects, as in Section 6.1, we briefly show a possible extension that allows the interactive counting system to better deal with anisotropic objects such as slightly elongated ones. The performance of the interactive system is demonstrated in Section 7.4 on a variety of microscopy images, as well as other visual material including satellite images, and finally, the summary and limitations are presented in Section 7.5.

7.1 Interactive System Overview

Given an image $\mathcal{I}$, the counting proceeds within a feedback loop. At each iteration, the user marks a certain portion of an image using a freehand selection tool (we refer to pixels being included into such regions as *annotated pixels*). Then the user *dots* the objects, by clicking once on each object of interest within this annotation region. In the first iteration, the user also marks a diameter of a typical object of an image by placing a line segment over an instance of the object.

At each iteration, given the set of dotted pixels $\mathcal{P}$ placed by the user on top of objects of interest in $\mathcal{I}$, our system aims to (1) build a codebook $\mathcal{X}$ of low-level features, (2) learn a mapping $\mathcal{F} : \mathcal{X} \mapsto \mathcal{Y}$ from the entries in the codebook $\mathcal{X}$ to an object density $\mathcal{Y}$, (3) use the learned mapping $\mathcal{F}$ to estimate the object density in the entire image $\mathcal{I}$, and (4) present the estimated object density map to the user through an intuitive visualization. The estimated object density map produced is such that integrating over a region of interest gives the estimated number of objects in it (e.g. integrating over the entire map gives an estimate of the total number of objects of interest in $\mathcal{I}$). By using the density visualization, the user can easily spot significant errors in the object density estimate, and can proceed to provide further annotations to refine the results in a next iteration of the process. An example with three iterations is shown in Figure 7.1. In practice, only a small portion or a few small portions of a potentially large image have to be inspected

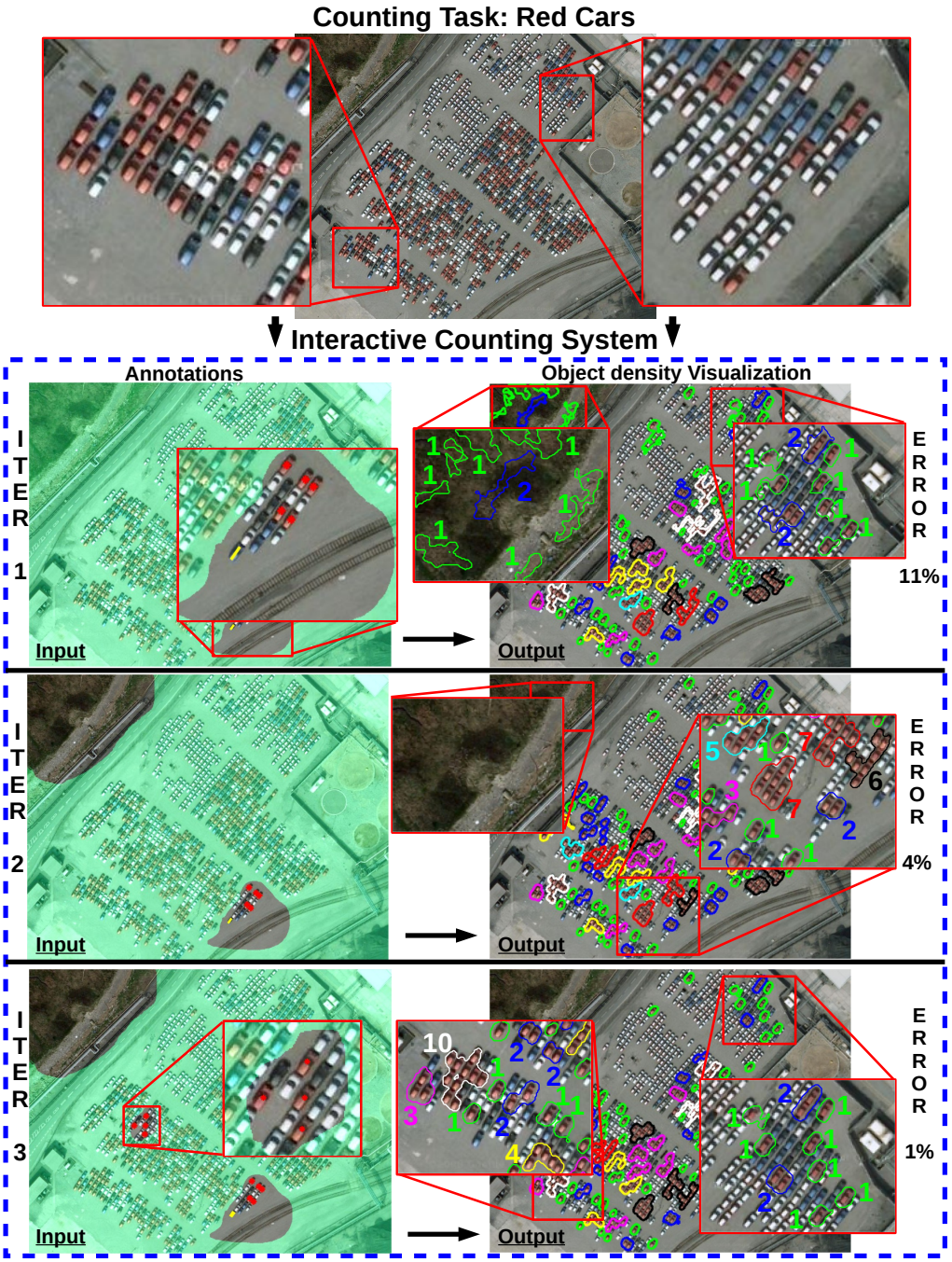


Figure 7.1: (Interactive Framework Overview) *Given an input image (or set of images) containing multiple instances of an object, our framework learns from regions with dot-annotations placed by the user in order to compute a map of* object density *for the non-annotated regions. The left column shows the annotated pixels provided by the user – all regions in the annotation images* outside *the green mask (i.e. with or without dot-annotations) are used as observations in the regression. The right column shown the intuitive visualization tool of the density estimation that allows the user to inspect the results and add further annotations where required in order to refine the output of the counting framework.*

in order to validate the correctness of the learned model or to identify parts with gross errors.

The first design requirement for an interactive counting system, such as the above, is that the codebook needs to be fast to compute, and adapt its size according to the amount of annotations in order to prevent severe under- or over-fitting. To address these problems, we propose in Section 7.2 a simple progressive codebook learning procedure, which builds a kd-tree that can grow from its current state as the user provides further annotations. The second requirement is a fast computation of the mapping $\mathcal{F}$. We make use of the pixel-level ridge regression presented in Section 6.2, which has a closed-form solution. Thus, the mapping $\mathcal{F}$ can be computed extremely fast through a few algebraic operations on sparse matrices. Finally, the system needs to present its current estimates to the user in such a way that identifying errors can be done through a quick visual inspection, which is not possible with the raw object density map and/or the global count. Therefore, we propose in Section 7.3 two methods to visualize object density maps by generating local "summaries" of it.

Our unoptimized MATLAB implementation takes at most a few seconds for each iteration of the relevance feedback which includes extending a codebook of visual features, discriminative (re)training, and visualization.

We show the performance of the interactive counting system through a series of examples in the experimental section, and videos of the interactive process are provided in [8].

7.2 Progressive codebook learning

Initially, we represent each pixel p of an image $\mathcal{I}$ with a d-dimensional real-valued vector $\mathbf{z_p} \in \mathbf{R}^d$. The idea of building a codebook on top of these low-level features is to subdivide the feature space into k cells, so that the typical density for appearance patterns falling into each cell is roughly constant. Ideally, we want to strike a balance between two conflicting goals. Firstly, we want to partition the feature space finely enough to avoid

under-fitting. Secondly, we want each of the partitions to have at least several pixels that belong to the area annotated by the user in order to avoid over-fitting. The latter requirement leads to the idea of interactive re-estimation of the feature space partition as more annotations become progressively available. This can be done very efficiently with the following algorithm.

We initialize the feature space partitioning by assigning all image pixels to the same partition. We then proceed recursively by splitting the partitions that contain more than N "annotated" pixels assigned to them ("annotated" here means belonging to the user-annotated area). Here, N is a meta-parameter selected by the user, which we set to 200 in our experiments. In more detail, the algorithms proceed as follows:

1. In the i-th iteration, find the partitions with more than N descriptors $\mathbf{z_p}$ assigned to it (only annotated pixels are taken into account).

2. For each of those partitions, find the feature dimension t of maximum variance (among the d dimensions), as well as the median of the values of all annotated pixels corresponding to this dimension.

3. Split such a partition into two according to whether a pixel value at the dimension t is greater or smaller then the median.

4. Repeat until every partition has less than N annotated pixels assigned to it.

The proposed algorithm thus constructs the kd-tree (w.r.t. the annotated pixels). Note, however, that we also maintain the partition assignments of the unannotated pixels (and the resulting partitions can be unbalanced w.r.t. the unannotated pixels). We finally note that there is no need to store the resulting kd-tree explicitly because the algorithm maintains the assignments of pixels to the leaves of the kd-tree (i.e. partitions). The partitioning algorithm is resumed whenever new annotations are added by the user. At this point, the codebook can grow from its current state by continuing the splitting (and is not re-learned from scratch).

Once the codebook has been learned, each pixel p in the image $\mathcal{I}$ is represented by a sparse k-dimensional vector x_p, where all entries are zero except the one corresponding to the partition to which the image descriptor z_p was assigned ("one-hot" encoding). The representation x_p is then used as the pixel features within the ridge-regression learning framework proposed in Chapter 6. Once again, we emphasize that the vector x_p changes (and becomes more high-dimensional) between the learning rounds as more user annotations become available.

7.3 Object density visualizations

The visualization of the object density estimate plays a key role in our interactive counting system as it assists the user to identify the parts of the image where the system has estimated the counts with large errors, and thus, where to add further annotations. While the predicted densities are sufficient to estimate the counts in any region, the accuracy of these densities cannot be controlled by the user without further post-processing due to the mismatch between the continuous nature of the densities and the discrete nature of the objects. To address this problem, we propose two density visualization methods, which convert the estimated density into representations that are intuitive for the user. The first method is based on non-overlapping extremal regions and is algorithmically similar to the detection method of Chapter 4, and aims to localize the objects from the density estimate. The second method is based on recursive image partitioning, and aims to split the image into a set of small regions where the number of objects can be easily eyeballed and compared to the density-based estimates.

For both visualization techniques, we start by generating a set of candidate regions with a nestedness property such that two regions R_i and R_j are either nested (i.e. $R_i \subset R_j$ or $R_j \subset R_i$) or they do not overlap (i.e. $R_i \cap R_j = \emptyset$). Therefore, the set of candidate regions in each case can be arranged into trees (as in Chapter 3 and Chapter 4). Both visualization approaches represent the densities by showing summations over a subset of those candidate regions and both approaches optimize the choice of this subset. In

particular, each region R_i has a score V_i associated to it that indicates the *integrality* of the region, or how well the region encloses an entire object or a cluster of objects. The idea is that we want to show the user the regions that have near-integer integrals of the density over them. Given the integrality scores, both approaches compute the final representation by picking a non-overlapping subset of regions that maximize the sum of such integrality-driven scores.

In more detail, we begin by defining S_i to be the integral of the estimated density map over the region R_i, and I_i to be the approximation of S_i to its nearest integer. The score V_i for region R_i is then defined as:

$$V_i = (1 - (S_i - I_i))^2 \tag{7.1}$$

For N candidate regions, we introduce the indicator variables $\mathbf{y} = \{y_1, y_2, ..., y_N\}$, where $y_i = 1$ implies that R_i has been selected. Additionally, $\mathbf{y}$ must satisfy the constraint of only containing non-overlapping regions. That is, $\mathbf{y} \in \mathcal{Y}$, where $\mathcal{Y}$ is the set of all sub-sets of non-overlapping regions such that if $R_i \cap R_j \neq \emptyset$ then $y_i.y_j = 0$.

The global maximization objective is defined as follows:

$$F(y) = \max_{y \in \mathcal{Y}} \sum_{i=1}^{N} y_i(V_i + \lambda) \tag{7.2}$$

where λ is a constant that prevents from selecting the trivial solution (one biggest region containing the whole image) and biases the solution towards a set of small regions. The objective (7.2) is optimized efficiently by using dynamic programming due to the tree structure of the regions as in Chapter 4. We now discuss the details of the two approaches and the difference between them.

Visualization using non-overlapping extremal regions. Extremal regions are the connected components on the binary images resulting from thresholding a gray image $\mathcal{I}$ with any arbitrary threshold τ. A key property of the extremal regions is the nestedness as described above. Therefore, the set of extremal regions of an image can be arranged

into a tree (or a forest) according to the nestedness.

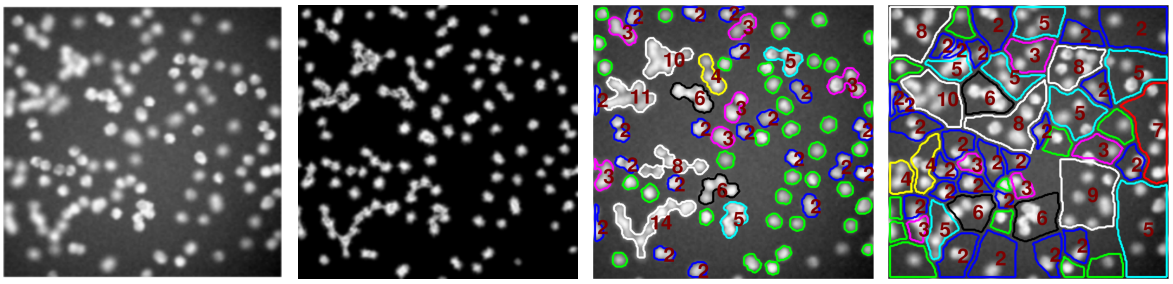
Similar to Chapter 3 and Chapter 4, we use extremal regions as candidates for object detection. In this case, however, extremal regions are extracted from the estimated object density map, and the ones selected by the optimization (7.2) should delineate entire objects or entire clusters of objects (Figure 7.2-c).

In practice, we collect these candidate regions using the method of Maximally Stable Extremal Regions (MSER) [104]. This method only keeps those extremal regions that are stable in the sense that they do not change abruptly between consecutive thresholds of the image (i.e. on regions with strong edges). During the inference, we exclude the regions which have an integral of density smaller than 0.5 from consideration as we have found that allowing any extremal region to be selected can result in very cluttered visualizations. Instead, this visualization aims to show only regions containing entire objects.

Visualization using hierarchical image partitioning. In this approach, we build a hierarchical image partition driven by the density. To obtain the partition, we iteratively apply spectral graph clustering, dividing image regions into two (akin to normalized cuts [138]). Unlike the extremal region visualization and unlike the traditional use of normalized cuts, we encourage the boundaries of this partition to go through regions of low density, thus creating a tile of regions that enclose entire objects (Figure 7.2-d). To achieve this, we build a 4-connected weighted graph $G = (V, E)$ with the adjacency matrix W defining the weights of the edges based on the estimated density map $F(p)$ as $w_{p,q}{=}0.5\,(F(p){+}F(q))$ for $(p, q){\in}\,E$.

The normalized cuts then tend to cut through the parts of the image where the density is near-zero, and also as usual have a bias towards equal-size partitions (which is desirable for our purpose).

Once the tree-structured graph is built, the inference selects the set of non-overlapping regions through the maximization of the sum of the integrality scores of the regions, as explained above. Additionally, we enforce at inference time that every pixel in the estimated density map must belong to one of the selected regions (i.e. that the selected



(a) *Input*

(b) *Density estimation*
Total count = 180

(c) *ER-visualization*
Total count = 163

(d) *SC-visualization*
Total count = 185

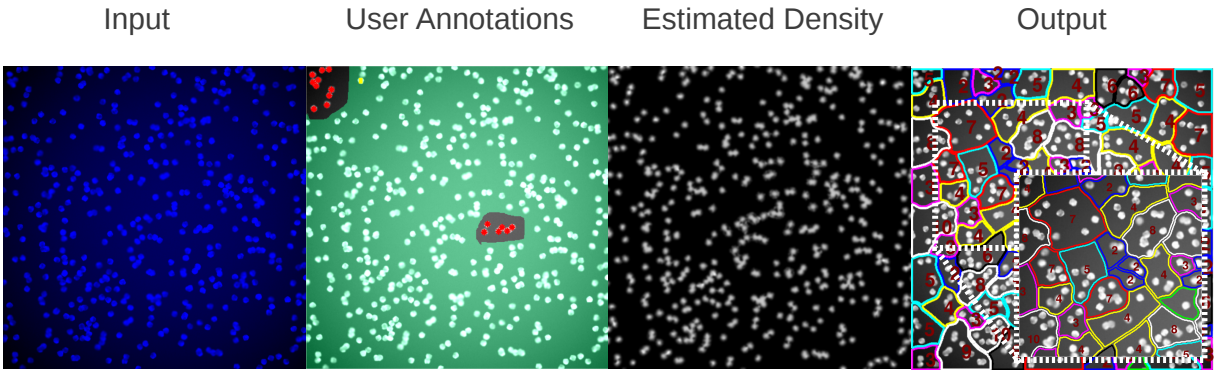
Figure 7.2: Density Visualization. *In order to assess the density estimation (b) of the original image (a), we propose two visualization methods. The first method (c) is based on non-overlapping extremal regions (ER) and aims to localize objects in the estimated density map (more intuitive but biased towards undercounting). The second method (d) is based on hierarchical image partitioning with spectral clustering (SC) and aims to explain the distribution of the density estimate across the entire image (higher fidelity but less intuitive visualization of the density). See text for details. In (c) and (d), the numbers indicate the objects contained within the region. Green regions contain a single object, but the number has been omitted for clarity. Non-outlined regions in (d) have zero counts.*

subset of regions represent a cover). Therefore, the entire density distribution is "explained". Accordingly, all regions from the hierarchical partitioning of the image are considered, including those with near zero density integrals.

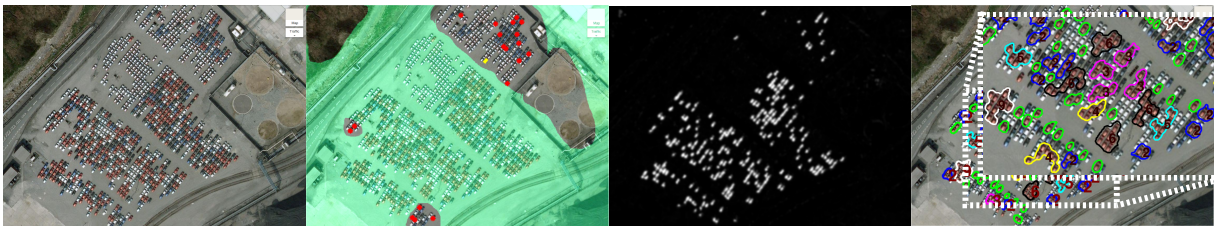
Compared to the visualization using the extremal regions, the visualization based on recursive partition does not tend to outline object boundaries, but represents the underlying ground truth density with greater fidelity due to the fact that the whole image ends up being covered by the selected regions (Figure 7.2).

7.4 Interactive counting experiments

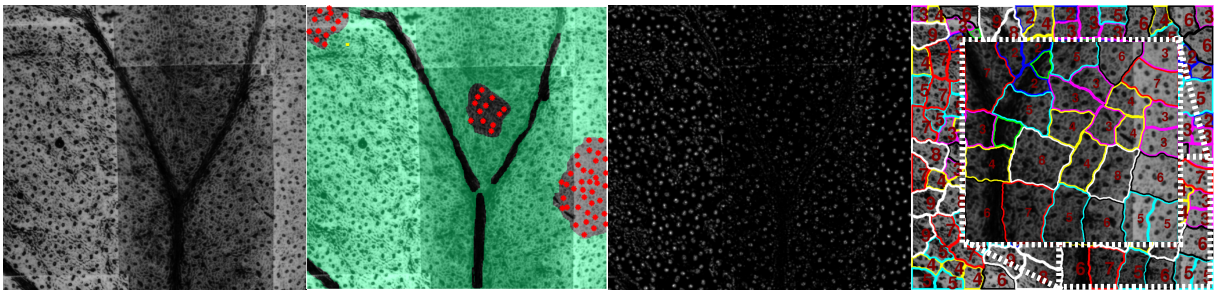
We now show the qualitative performance of the interactive counting system in Figures 7.3, 7.4, 7.5 and 7.6. All figures show example results of the interactive counting system, indicating the number of annotations added and the estimated object count for that amount of annotation. The aim is to give a sense of the amount of annotations (and effort) required to obtain an object count that would closely approximate the ground truth (i.e. with an approximate absolute counting error of 10% or less). This section is



(a) Synthetic cells. Number of dot-annotations = 16. Estimated count/GT = 476/484. Reference results from [159] = 482/500.

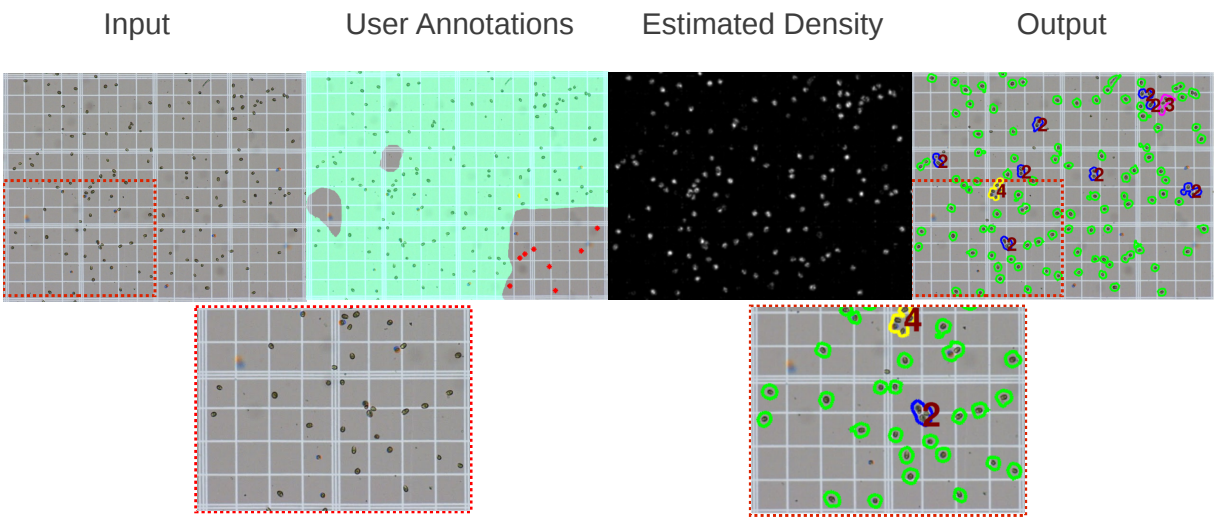


(b) Red cars. Number of dot-annotations = 19. Estimated count/GT = 220/230.

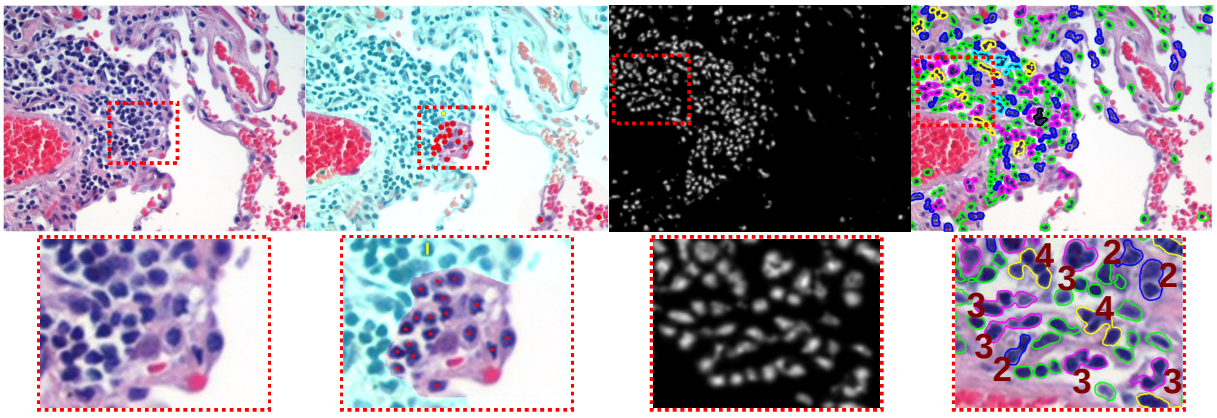


(c) Stomata. Number of dot-annotations = 37. Estimated count/GT = 655/676. Reference results from [159] = 716/676.

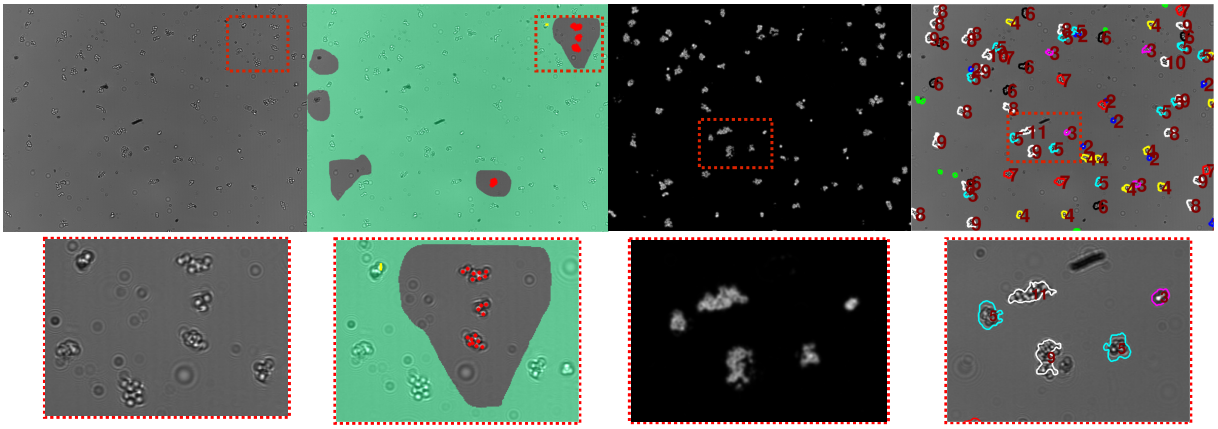
Figure 7.3: Example results. *A large image (first column) is annotated interactively (second column) until qualitatively reasonable results are produced (fourth column). The visualization of the results (fourth column) is computed from the estimated density map (third column) as described in Section 7.3. See Section 7.4 for details.*



(a) Counting Cells on Hemocytometer Number of dot-annotations = 8. GT/Estimated count = 123/124.

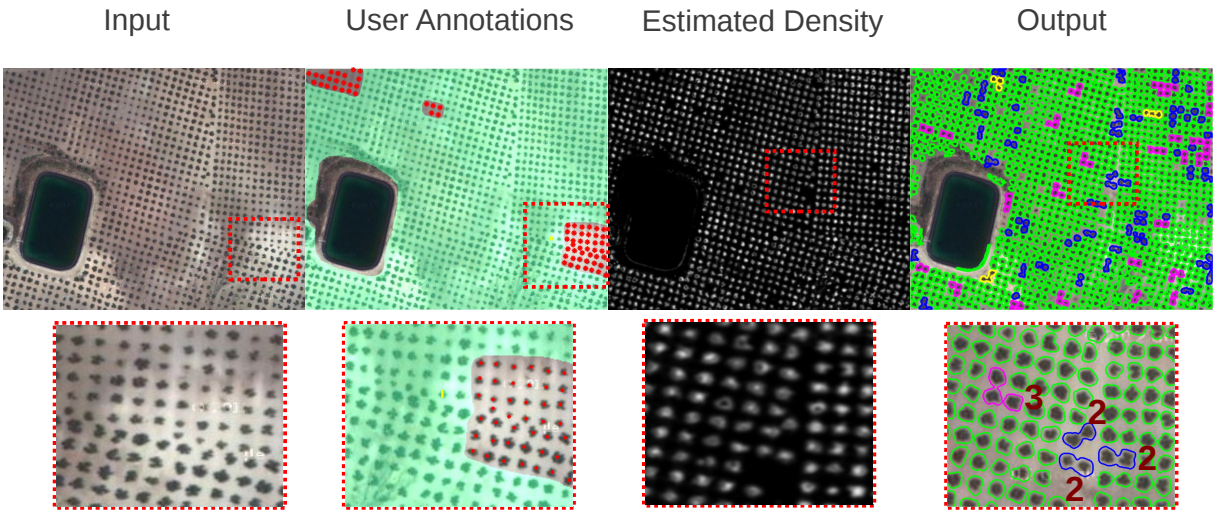


(b) Counting Cells in Histology Number of dot-annotations = 20. GT/Estimated count = 384/387.

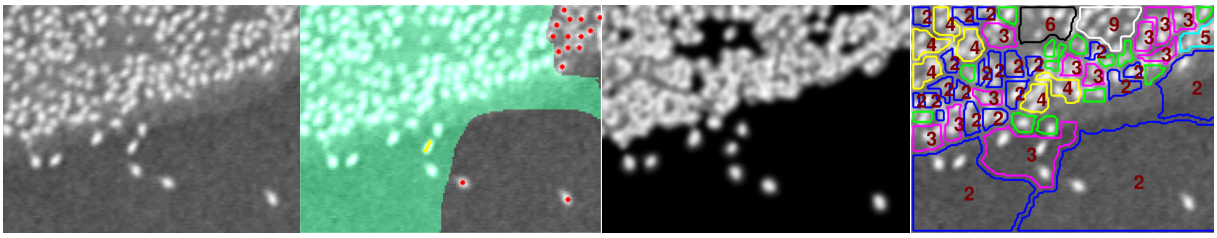


(c) Counting Cells in Lens-free and Large Field-of-View Microscopy Number of dot-annotations = 20. GT/Estimated count = 478/468.

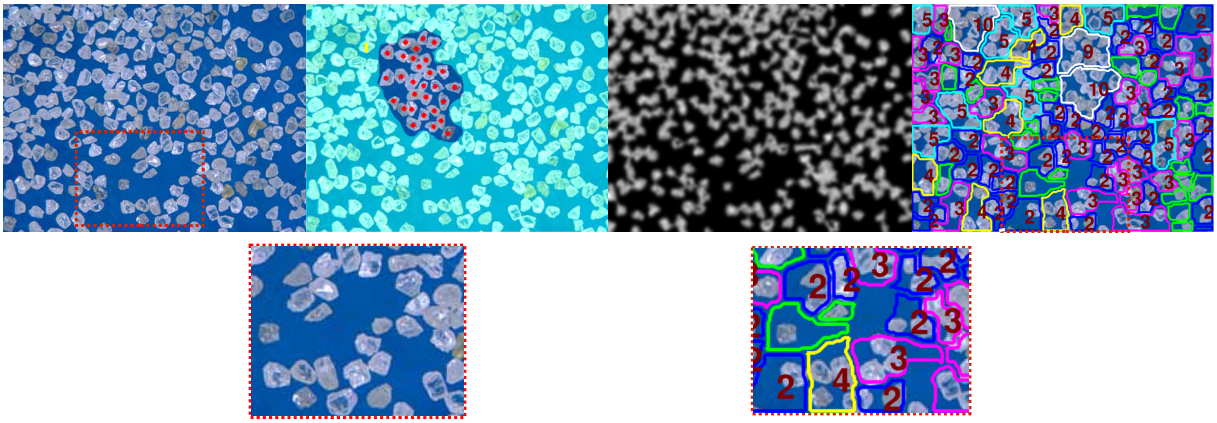
Figure 7.4: Additional examples on microscopy images. *A large image (first column) is annotated interactively (second column) until qualitatively reasonable results are produced (fourth column). The visualization of the results (fourth column) is computed from the estimated density map (third column) as described in Section 7.3. See Section 7.4 for details.*



(a) Counting Olive Trees Number of dot-annotations = 68. GT/Estimated count = 1165/1239.

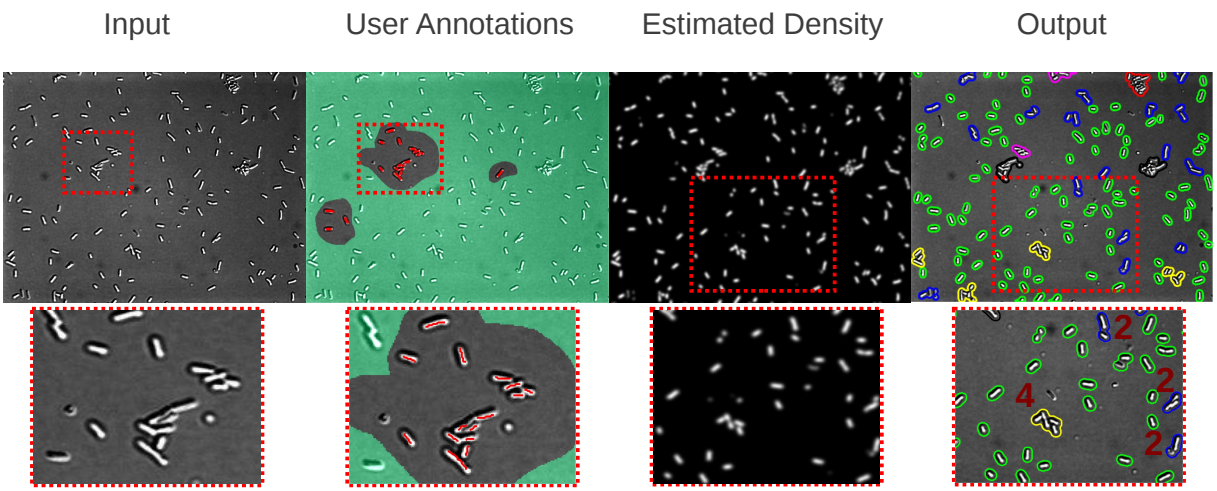


(b) Counting Birds in a Colony Number of dot-annotations = 15. GT/Estimated count = 127/128.



(c) Counting Diamonds Number of dot-annotations = 29. GT/Estimated count = 250/258.

Figure 7.5: Additional examples on aerial images and other objects. *A large image (first column) is annotated interactively (second column) until qualitatively reasonable results are produced (fourth column). The visualization of the results (fourth column) is computed from the estimated density map (third column) as described in Section 7.3. See Section 7.4 for details.*



(b) Counting ecoli in microscopy image. Number of stroke-annotations = 19. GT/Estimated count = 157/162.

Figure 7.6: Counting elongated objects *A large image (first column) is annotated interactively (second column) until qualitatively reasonable results are produced (fourth column). The visualization of the results (fourth column) is computed from the estimated density map (third column) as described in Section 7.3. In this case, due to the elongated shape of the object of interest, we use stroke annotations instead of dots. This type of annotations allows us to handle anisotropic objects, as described in Section 7.4.1.*

complemented with [8], where a video of the system in use is shown. Note that, even though results are shown here using the same images as input and output, it is possible to propagate the density estimation to other similar images in a batch.

For all examples, the green masks on the annotation images (second column) indicate regions that have not been annotated by the user. All regions outside the green mask, with or without dot (or stroke) annotations, are used as observations in the regression (Section 6.2). Annotated regions without dots or strokes can be seen as zero annotations. As expected, the number of annotations required increases with the difficulty of the problem. In cases where the background is complex, such as in aerial images, residual density tends to appear all over the image, which can be seen in the visualization. However, this can be easily fixed interactively using zero annotations, which are very fast and simple to add.

Some of the examples (Figures 7.3-a,c) have been taken from the benchmark dataset of [159], and we show their results *as reference* in Figure 7.3. We do not attempt to do a direct comparison of performance with [159] due to the fact that for the cases where a

single image is given, our interactive method requires annotations on this image in order to produce results, and thus, disrupts the possibility of a fair comparison of performance. Moreover, due to the nature of the low-level features, our system crops the borders of the image by half the size of the texture patches (see implementation details), resulting in a possible difference of the ground truth count w.r.t. the original image. The additional examples correspond to various microscopy images from Google image search, or aerial images extracted from Google Maps.

The same set of parameters have been used for all the examples shown, with the most relevant ones indicated in the implementation details. We show a single visualization method for each of the examples, but it can be seen that they are complementary. Nevertheless, depending on the image, one visualization can be more convenient than the other.

7.4.1 Implementation details

Low-level features. We compute the initial (low-level) pixel descriptor $\mathbf{z_p}$ based on two types of local features on the *Lab* color-space. First, we use the contrast-normalized lightness values (L channel) of the pixels in a patch of size $n \times n$ centered at p [157]. The patches are rotated such that their dominant gradients are aligned in order to be invariant to the object's rotation in the image. Secondly, we collect the raw L, a and b values of the center pixel. The descriptor $\mathbf{z_p} \in \mathbf{R}^d$ is the concatenation of the two local features. Therefore, the dimensionality d of the pixel descriptor for a color image is $n^2 + 3$. In the case of grayscale images, we do the feature computation on the given intensity channel, which results in $d = n^2 + 1$.

Collecting extremal regions. Extremal regions are extracted from the estimated density map using the MSER implementation from *VLFeat* [158]. In order to collect enough candidate regions for the inference to select from, we set a low stability threshold in the MSER algorithm.

Building a binary tree with spectral clustering. Computing the traditional spectral

clustering as in [138] can be too slow for our interactive application, and the reason is the expensive computation of eigenvectors. Therefore, in practice we use the method from Dhillon *et al.* [47] which solves the equivalent problem of weighted kernel k-means thus greatly reducing the computation time. We use the implementation from the authors of [47].

Setting object-size dependent parameters. Some of the parameters used in the implementation of the interactive counting system are better set with respect to the size of the object of interest. These are the size $n \times n$ of the patches for the low level features and the standard deviation σ for the Gaussian kernel used to smooth the dot-annotations and feature channels for the ridge regression. As discussed in Section 7.1, we chose to request an additional input from the user, where the approximate diameter of the object of interest is input by drawing a line segment over a single object in the image. The image is then rescaled with the scale factor of the object. For the experiments of the interactive system shown in the experimental section, we use an object size of 10 pixels, patches of 9×9 pixels and $\sigma = 3$ pixels.

Region visualization. The boundaries of the regions that are chosen to visualize the density are superimposed on top of the original images. Alongside the boundaries, we show the density integrals over the highlighted regions rounded to the nearest integer (recall that regions are chosen so that such integrals tend to be near integer). We also colour-code the boundaries according to the counts (e.g. green for objects containing one object, blue for two objects, etc.).

Stroke annotations. Through the counting chapters we have used dot-annotations as the main source of user input for the density estimation task. Nevertheless, in the case of anisotropic objects, the dot-annotations can lead to poor ground-truth density maps which can significantly affect the learning when only few annotations are provided (e.g interactive counting). For such cases we use a simple variant of the annotation type in the form of stroke-annotations, which is handled in the same way as the dots but, prior to the Gaussian smoothing, the density is distributed over the path of the stroke as opposed

to a single pixel. Additionally, provided that the objects are of similar size, the stroke annotation can directly provide the size indication that was previously obtained through the diameter drawing tool. An example of the stroke-annotation variant is shown in Figure 7.6 in order to interactively count elongated bacteria in a microscopy image.

7.5 Summary and limitations

This chapter presented a first foray into enabling counting, previously treated as a traditional batch learning problem, to be handled interactively. To do this we have proposed a solution that builds upon on-the-fly learning of object densities and overcomes the challenge of efficient density visualization. The result is an agile and flexible system which enables quite disparate visual material (spanning both microscopy images of cells and satellite imagery) to be annotated and counted in a matter of seconds.

There is certainly room for improvement: firstly, the features used can be extended to enable more local and contextual information to be captured. Secondly, our current system does not handle perspective geometry and cannot be directly applied to images with objects on a slanted ground plane. The latter, however, can easily be fixed by allowing a projective transformation to be imported or by the user providing additional object size annotations.

Chapter 8: Exploring image-based HTS

In this chapter we address the exploration of microscopy datasets coming from large exploratory studies by providing a pipeline for discovering and visualizing the effects over a target of interest.

We consider the scenario of an image dataset that would typically come from exploratory studies collected on high-throughput screening (HTS) platforms, where a large set (thousands or millions) of external perturbations such as small-molecules or RNAi, collectively known as perturbagens, are applied to a set of samples (e.g. wells containing cell cultures) which are then imaged to visualize the effects on a target (e.g. a cell protein that has been fluorescently tagged). In such scenario, the goal of the methods presented in this chapter is to aid in the analysis of the exploratory study by discovering what the visual patterns are that the set of perturbagens create over the target, which compounds or groups of compounds create each of the possible visual effects, and what characteristic of the perturbagens could have led to them. In order to provide the answer to those questions, we define three specific tasks for the vision-based method. The *first task* consists on displaying in a simple manner (e.g. on two dimensions) the different clusters of visual patterns that can be found throughout the dataset; the *second task* is a visualization method to highlight the patterns that distinguish one cluster from another, and the *third task* is a method to relate those visual patterns to properties in the perturbagens in order to generate hypotheses about the underlying mechanisms.

This chapter is structured in the following way. In Section 8.1, we describe the datasets used in our experiments, followed by an overview of our exploration tool in Section 8.2. Then we present a section for each of the three tasks that we aim to achieve, as described

above. Finally, a summary and a discussion of the limitations are presented in Section 8.6.

8.1 HTS datasets

We demonstrate the usage of the HTS exploration tools on two datasets coming from a small-molecule screening on the *apoptosis-stimulating of p53 protein 2* (ASPP2), aiming to explore the effects of clinically approved compounds over junctional ASPP2. The ASPP family of proteins play an important role in cell apoptosis, and the failure of this process has been linked to the development of cancer. It is therefore of great interest to understand the mechanisms behind the expression of the ASPP proteins.

The datasets used in our experiments consist of i) the screening of the Pharmakon 1600 library (1600 compounds - Microsource Discovery Systems, Gaylordsville, CT, USA) and ii) the screening of the SGC library (2000 compounds - SGC Laboratories), both at 20uMol on the Caco-2 colorectal cancer cell line. The dataset is part of a larger study being performed by Jaroslav Zak and Dr. Xin Lu at the Ludwig Institute of Cancer Research, University of Oxford, in collaboration with the Target Discovery Institute, Nuffield Department of Medicine, University of Oxford.

In both datasets, the screening was performed in 96-well plates, with the library compounds, positive and negative controls arranged in the pattern shown in Figure 8.1. The experiments were performed with duplicates (i.e. every plate was identically prepared and imaged twice). The control dishes have a known behaviour, thus are used to ensure that the experiments are being performed on the expected conditions, as well as baselines to assess the extent of the effects caused by the library compounds. In the ASPP2 datasets, the negative controls correspond to wells where no compound is applied and the protein can be seen in its normal state, which corresponds to the cell tight junctions in the case of the Caco-2 cell line. Likewise, the positive controls correspond to wells treated with a compound known to completely disrupt the junctional state of the ASPP2 protein.

Within each well, 7 fields of view (i.e. regions within the well) were imaged at two

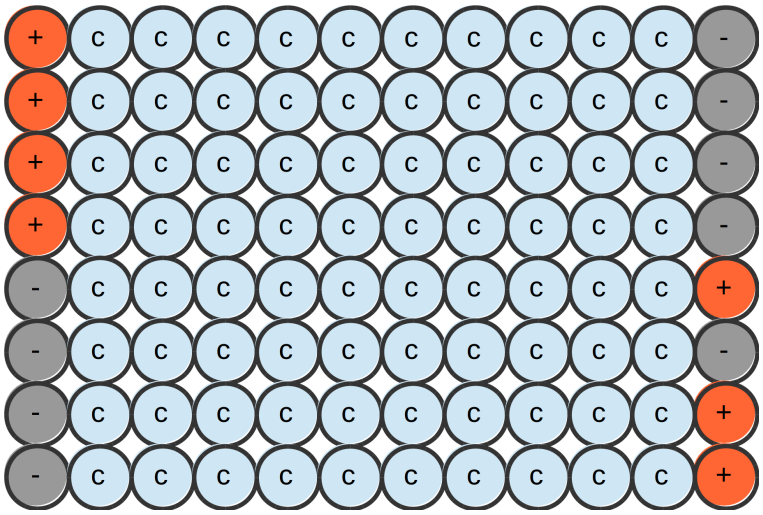


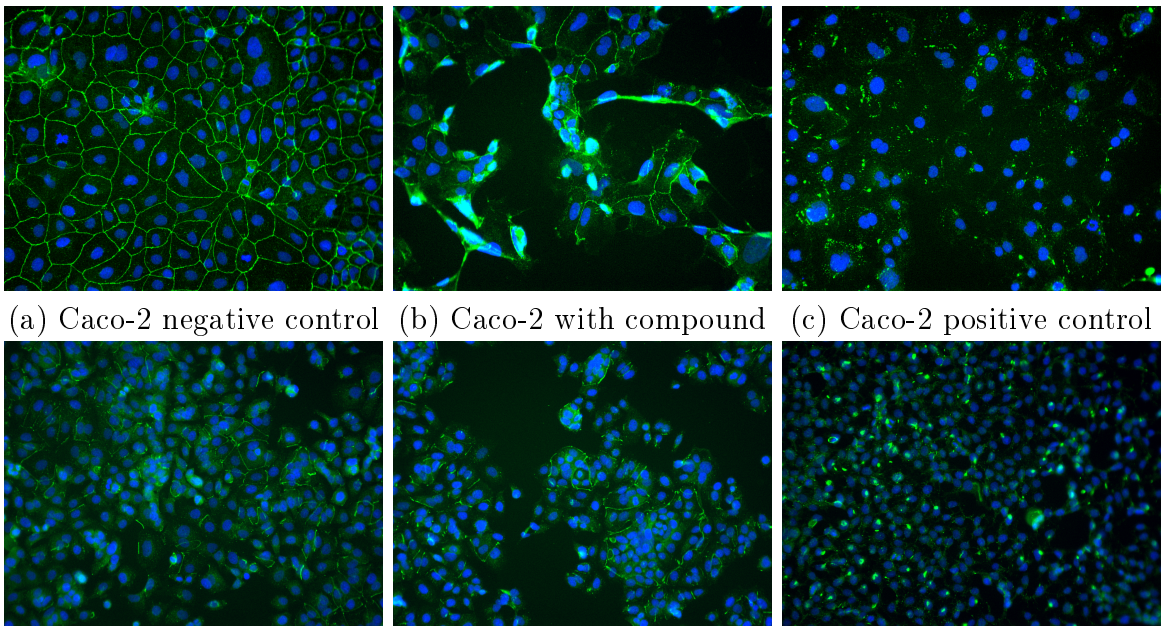
Figure 8.1: *Diagram of the arrangement of compounds on the plates used in the screenings described in Section 8.1. The plates consist of 96-wells, where positive (+) and negative (-) controls are placed in the first and last column, and the remaining wells are treated with the compounds (C) of the respective libraries used in the screenings.*

different wavelengths, thus producing two image channels: the cell nuclei and the ASPP2 fluorescent marker. A few example fields of view on normalized images[1] can be seen in Figure 8.2, where the blue and green channel correspond to the cell nuclei marker and ASPP2 marker respectively. We refer to the collection of the 7 fields of view taken from the same well as a single sample. The image channels were collected at 1 Megapixel with a 16-bit resolution.

8.2 Overview

In order to represent the visual similarities within a dataset in an intuitive manner (*task i*), we aim to display the samples in a 2-dimensional plot (Figure 8.3a) such that samples close to each other are "more similar" in terms of their visual features than samples that are further apart. Therefore, it is expected that a small neighbourhood in the 2-D space would contain samples where the effect over the target is visually similar (Figure 8.3b). Furthermore, depending on the application, the similarities could be required in more

[1]The image normalization consists on limiting the range of the channels and converting them into 8-bits in order to visualize the samples. This procedure is only done for visualization purposes, as the computations described later are performed on the raw images.



(a) Caco-2 negative control (b) Caco-2 with compound (c) Caco-2 positive control

(d) MCF-7 negative control (e) MCF-7 with compound (f) MCF-7 positive control

Figure 8.2: *Example images from the ASPP2 screenings. A negative control (first column), compound well (second column) and a positive control (third column) are shown for the Caco-2 cell line (top row) and on MCF-7 cell lines (bottom row). The blue channel corresponds to a fluorescence marker tagging the cell nuclei, whereas the green channels tags the ASPP2.*

or less detail, for which we compute multiple 2-D maps at different levels of detail (Figure 8.3c) that the user can navigate through. An additional example can be seen in Figure 8.4.

Given the 2-D maps of sample similarities, it is important to have the possibility of visualizing objectively the specific patterns within a neighbourhood that cause the samples to be considered similar (*task ii*). Therefore, we present a tool allows the user to specify a neighbourhood of interest where the image regions containing the distinctive patterns of the neighbourhood are then highlighted (Figure 8.6).

Finally, given a neighbourhood with a specific visual pattern of interest (e.g. translocation of the target), it is possible to compute the likelihood that the accumulation of certain compound property within the neighbourhood containing the pattern does not happen by chance, known as enrichment (*task iii*). That is, if a neighbourhood is "enriched" with compounds that contain the property z (e.g. the therapeutic use of the compounds), then it is likely that such property is related to the formation of the visual

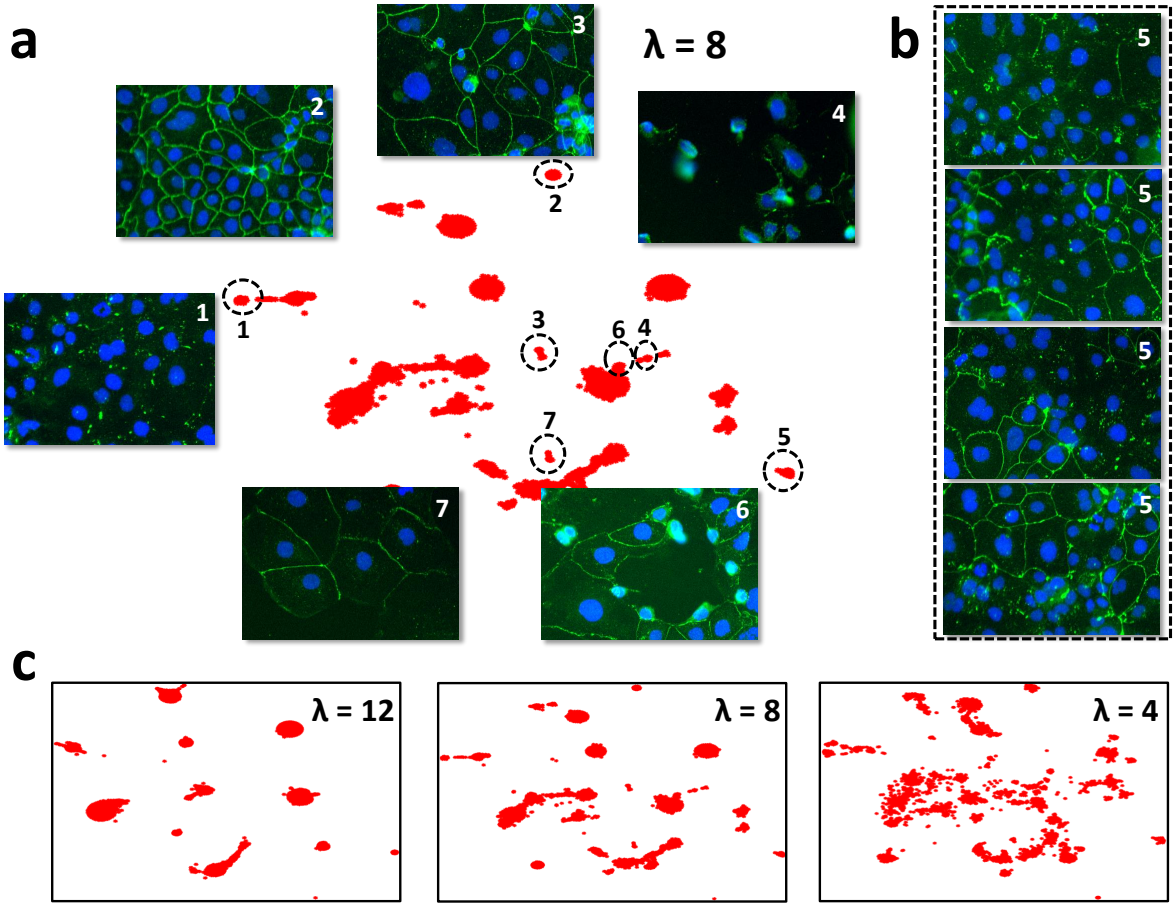


Figure 8.3: *Visualizing a high-throughput screening according to the visual similarities in its samples. In this example we show the ASPP2 screening on the Caco-2 cell line with the Pharmakon 1600 library at 20uMol.* **(a)** *Every sample (red asterisk) in the data set is projected into a 2-D space where samples laying near to each other are considered visually similar. Therefore, exploring the different naturally-formed clusters in the 2-D space gives an idea of the different range of visual effects in the experiment, as different clusters are expected to contain different visual appereances as shown for clusters 1-7.* **(b)** *Samples lying within the same cluster are expected to present the similar visual patterns.* **(c)** *The level of detail considered in the 2-D visualization can be varied with a parameter λ, resulting in finer of coarser visualizations of the dataset; $\lambda = 8$ was used for* **(a)**.

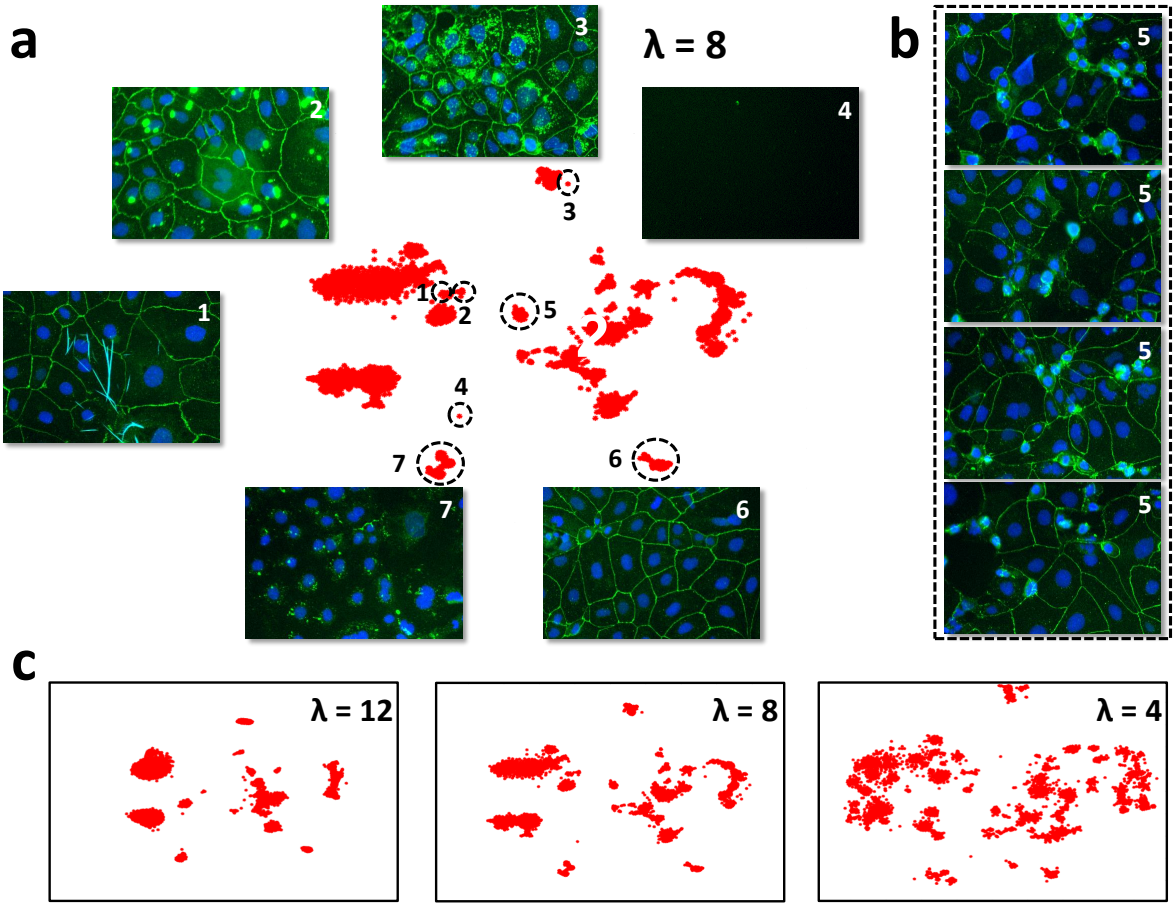


Figure 8.4: *Visualizing a high-throughput screening according to the visual similarities in its samples. In this example we show the ASPP2 screening on the Caco-2 cell line with the SGC library at 20uMol.* **(a)** *Every sample (red asterisk) in the data set is projected into a 2-D space where samples laying near to each other are considered visually similar. Therefore, exploring the different naturally-formed clusters in the 2-D space gives an idea of the different range of visual effects in the experiment, as different clusters are expected to contain different visual appereances as shown for clusters 1-7.* **(b)** *Samples lying within the same cluster are expected to present the similar visual patterns.* **(c)** *The level of detail considered in the 2-D visualization can be varied with a parameter λ, resulting in finer of coarser visualizations of the dataset; $\lambda = 8$ was used for* **(a)**. *In this second visualization example, clusters 1 and 2 contain experimental artifacts, and cluster 4 contain compounds that were highly toxic for the cell line. Even though such examples are not interesting from the biological perspective of the experiment, we intentionally show them as they contain visual patterns that are easy to visualize.*

pattern that led to the samples being visually similar. Computing enrichments can be useful in order to formulate hypotheses regarding the relation between the visual pattern and the compound property (Figure 8.7). The three tasks are detailed next.

8.3 Measuring and visualizing sample similarities

The core of the HTS exploration tool is the capability of quantifying general visual similarities between images in an unbiased manner, that is, without the necessity of predefining the specific visual effects that should be taken into account. This is achieved through the application of Fisher Vectors [121] for image-level encoding of local image features: each sample in the database is represented by a single vector that encodes the local and global patterns in the images corresponding to a single sample, which are extracted from the channel of interest, in this case, the ASPP2 image channel (i.e. the green channel in Figure 8.2).

8.3.1 Image encoding

The first step in the Fisher Vector-based sample encoding consists in collecting local visual features from the images of each sample. Following recent computer vision literature [31, 140] for image level classification, we use the 128-dimensional SIFT [96] descriptors sampled densely (every two pixels) and at five different scales over the channel of interest (e.g. the ASPP2 channel in the Caco-2 data set). The dimensionality of the SIFT descriptors is reduced from 128 to 80 via PCA for decorrelation and then square-rooted, as it has been shown to improve their performance for image similarity measures [5]. The pool of descriptors from the dataset is randomly sampled, keeping a smaller but representative set of descriptors that is used to fit a Gaussian Mixture Model (GMM) with 128 components that will represent the space of “visual words" throughout the dataset. Given the GMM, the code for each sample in the dataset is built by accumulating the first and second order statistics of its local descriptors (multi-scale 80-dimensional SIFT) w.r.t. each component in the GMM; this is known as Fisher encoding [121].

As shown in [31], the whole image encoding can benefit from some spatial information, as this is not retained by default on the Fisher encoding. Therefore, spatial pooling is used for natural images, as illustrated in Figure 2.11. In the case of microscopy images, we provide spatial information by creating separate encodings for the image regions that lie inside and outside of the cell nuclei, and the two separate codes are then concatenated into a single vector. In order to determine the different regions of interest for the cell experiments using the cell nuclei as reference, it is necessary to segment the nuclei from each image. Because of the nature of the experiments, it is straightforward to obtain nuclei and non-nuclei regions by applying a simple global thresholding of the nuclei channel. Such procedure can be extended if the screening provides further image channels to use as reference such as other sub-cellular organelles. In practice, we maintain the total number of components in the GMM (and thus the dimensionality of the codes) as we divide the number of components by the number of separate codes that will be built. For instances, in the ASPP2 screening we use a GMM of 64 components for each of the two regions of interest.

Considering the 80-dimensional low-level descriptors, and the 128 components of the GMMs, the resulting code for each sample is 20480-dimensional. Each code x_i is transformed as $sign(x_i)|x_i|^{1/2}$ and then L_2-normalized. When the sample consists of several images (e.g. 7 fields of view in the case of the ASPP2 screening), each image is encoded separately and the codes are summed before the transformation and normalization.

The result of the encoding procedure is a single high-dimensional feature vector x_i for each sample in the database, which can then be used to determine how visually similar (or dissimilar) two samples are (e.g. using euclidean distance between vectors $\|x_i - x_j\|$), and build a matrix of pairwise distances for the entire dataset. An example of the distance matrix computed for the dataset in Figure 8.3 is shown in Figure 8.5.

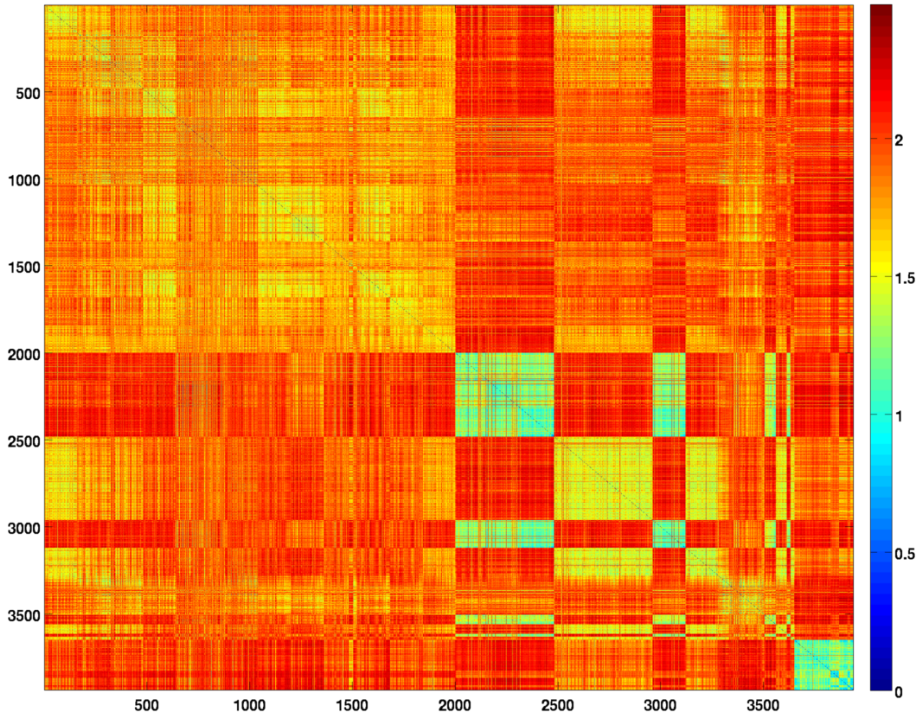


Figure 8.5: *Example of the Euclidean distance matrix computed for the dataset that is being visualized in Figure 8.3. See text for details.*

8.3.2 2-D dataset visualization

Given a matrix D containing the distance between every pair of samples s_i and s_j in the data set, we wish to present all the relations within it to the user in an intuitive way. Therefore, we aim to project the samples into a 2-dimensional space that can be easily interpreted. However, the low-dimensional projection of high-dimensional data often comes with a compromise on the accuracy of the representation. In order to alleviate this problem, we make use of the t-distributed stochastic embedding (t-SNE) method of Van der Maaten and Hinton [155], a nonlinear dimensionality reduction method specially developed for the visualization of high-dimensional data, which was described in Section 2.4. t-SNE is capable of capturing the structures in a manifold where the high-dimensional data lies. Therefore, by emphasizing local or global structures in the manifold, it is possible to obtain more or less detailed visualizations that might be more appropriate depending on the application. Such a goal can be achieved by propagating the similarities through the high-dimensional manifold at different extents using diffusion processes

[50]. Intuitively, the further the similarities are propagated, the smoother the manifold is perceived. In practice, the propagation is done on an affinity graph built from D, where every sample i is a node, and the weights between nodes i and j is given by their distance $D_{i,j}$; the similarities are then diffused from each node to its λ-nearest neighbours, and thus, λ controls the extent of the diffusion process. We compute the t-SNE 2-D visualization over the diffused distance matrix for various values of λ, successively initializing the t-SNE optimization from the previous t-SNE output in order to maintain the spatial consistency between samples[2], generating visualizations at different levels of detail that the user can navigate through (Figure 8.3c).

8.4 Highlighting cluster similarities

The 2-D visualizations (Figure 8.3 and Figure 8.4) are generally populated with naturally-formed clusters of similar images. Nevertheless, it is not always clear what the specific patterns are that determine the similarities between the samples in a cluster, e.g. the samples on Figure 8.3b. Therefore, we aim to produce, for each image that belongs to some cluster of interest C_k, a highlighting map that displays the relative contribution of each of the local image features towards the formation of the cluster, as shown in Figure 8.6.

We begin by learning a weight vector w through an SVM, where the Fisher Vectors of the samples in C_k are used a positive examples, and the Fisher Vectors for the rest of the samples are used as negatives. Then, on a test image (i.e. an image from a sample in C_k where we want to visualize the discriminative patterns), we apply the Fisher encoding procedure to each pixel p. Finally, the highlighting map is created by assigning to the location of p the dot product of the encoding of p and the vector w. Intuitively, the learning procedure would determine the modes of the GMM that are particular to C_k and in what degree of importance. Therefore, every pixel p in the highlight image can

[2]The cost function in t-SNE is non-convex. Therefore, different local minima can be reached when using different initializations.

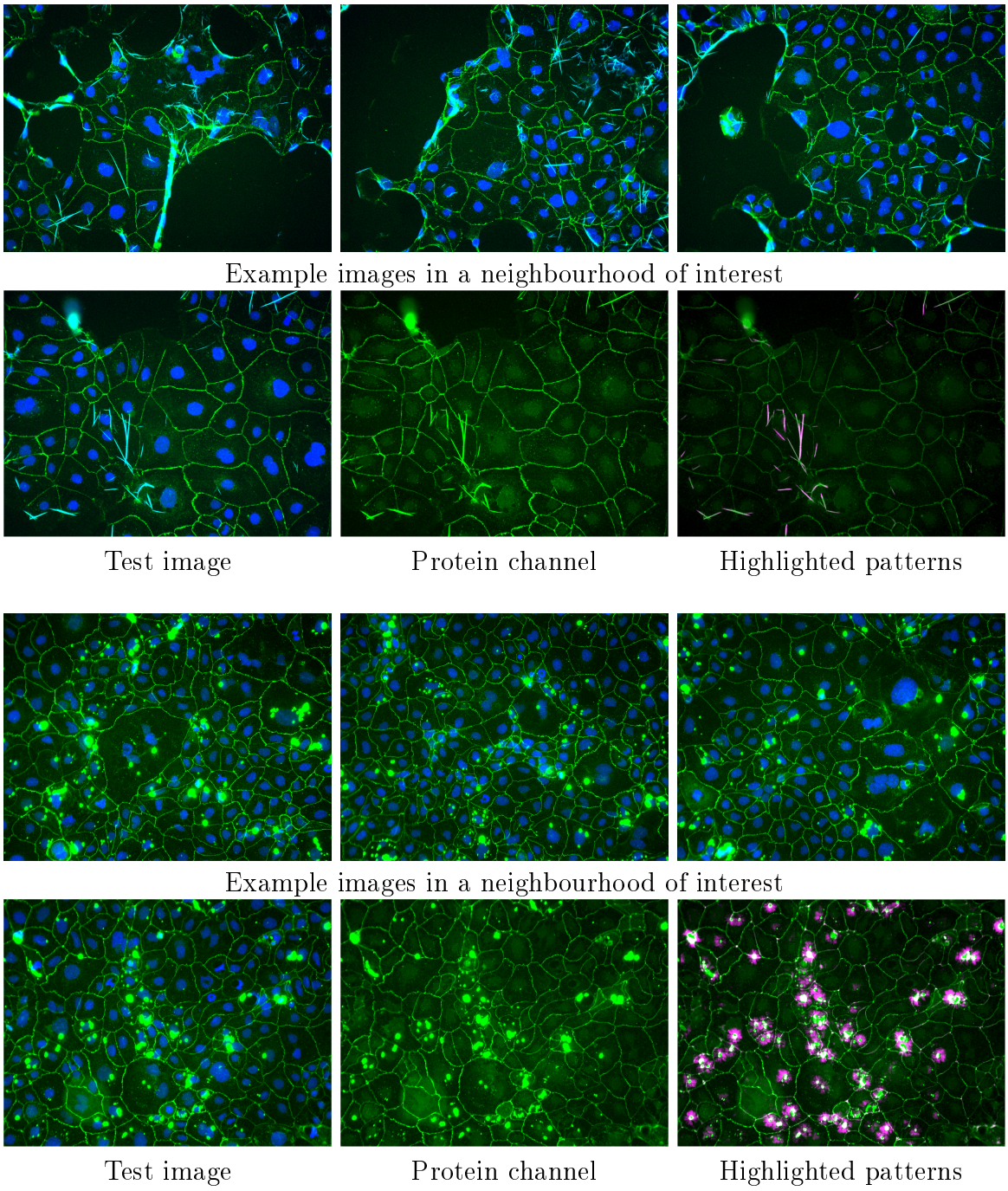


Figure 8.6: *Two examples of highlighting discriminative visual patterns of neighbourhoods in the 2D visualizations by using the technique described in Section 8.4. Both examples are taken from the 2D visualization of Figure 8.4, corresponding to neighbourhoods 1 (*top two rows*) and 2 (*bottom two rows*). Within each of the two examples, the upper row shows some of the sample images found in the neighbourhood of interest, and the lower row shows a test sample in such neighbourhood (*left*), where the highlighted patterns are shown in magenta (*right*). For the clarity of this figure, neighbourhoods with unambiguous visual patterns were been chosen in order to illustrate the output of the highlighting method. Nevertheless, these examples correspond to experimental artifacts which are not necessarily relevant from the biological perspective of the ASPP2 screening.*

be assigned the weight of the GMM mode under which the visual features around p are more likely, thus revealing the importance to C_k of the different visual patterns in the images.

In general, the highlighting can be done over some neighbourhood of interest N_{s_i} (i.e. the neighbourhood around the sample i), without the necessity of producing an automatic hard-clustering of the samples. For example, an interesting neighbourhood can be one that appears isolated in the 2D visualization. In this case, the user might be interested in objectively visualizing the patterns in the isolated cluster that the algorithm found to be be discriminative, and the selection of such cluster can be done by the user through a graphical interface. In Figure 8.6, we show examples of well-defined patterns from the SGC on Caco2 screening which were selected from isolated clusters in Figure 8.4a. We intentionally choose visually striking clusters regardless of their biological meaning for the purpose of presenting the output of the highlighting method on unambiguous cases.

Alternatively, a neighbourhood of interest can be one that is enriched with a property of the compounds which can be found automatically within the screening exploration pipeline. In this case, metadata of the compounds can be used to link clusters in the 2D visualization with non-random concentration of compounds with certain property in common such as the chemical structure of the compounds, or their therapeutic use. An example of the latter is shown in Figure 8.7, and the process to find enriched neighbourhoods is described next.

8.5 Finding enrichments

When computing the enrichment of a property on the naturally-formed clusters (as explained below), we make a distinction between the clusters in the 2-D visualization, and the neighbourhoods in the high-dimensional space. Indeed, projecting relative distances of high-dimensional vectors into a 2-D space will carry representation errors. Therefore, when defining the samples that lie next to some sample s_i (the neighbourhood N_{s_i}), we consider the samples that lie within a distance ρ of s_i in the high-dimensional space, and

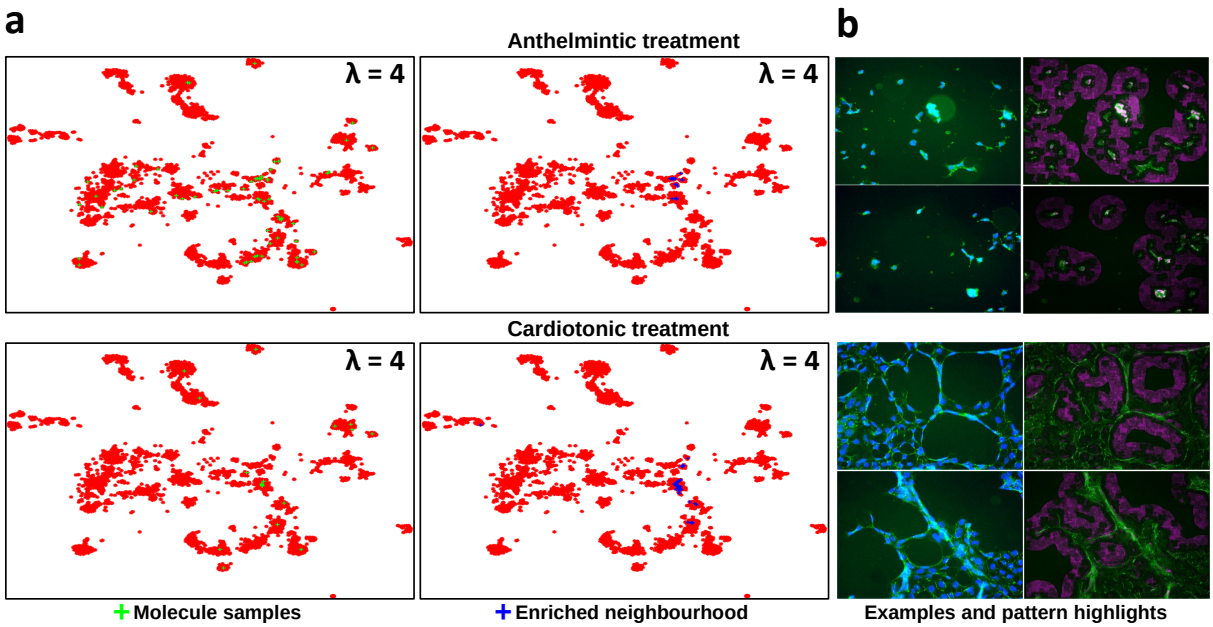


Figure 8.7: *The different neighbourhoods formed according to the visual similarities between samples can be evaluated for enrichment of certain experimental variable. In this example, we search for neighbourhoods enriched according to the therapeutic use of the molecules used to treat the samples, where enrichments for anthelmintic (top row) and cardiotonic (bottom row) drugs were found ($p<0.01$).* **(a)** *The first column highlights, in the 2D visualization, the different samples that contain the property of interest (i.e. different compounds with a commom therapeutic use), whereas the second column highlights the neighbourhoods found to be enriched with such property.* **(b)** *Using the visual pattern highlight tool, we are able to visualize the image regions that present the most distinctive visual patterns in the enriched neighbourhoods.*

not in the 2-D visualization.

The naturally-formed clusters of samples can also be used to formulate hypotheses about relations between visual patterns and some experimental variables, which can be visualized on the 2-D representation of Figure 8.3a. In order to do this task, we first define a neighbourhood N_s for every sample s in the dataset. Each neighbourhood N_s is then used to evaluate the probability that the accumulation of samples with the property of interest happens by chance; if not, a relation between the visual patterns of the samples in N_s and the property of interest can be hypothesized.

To demonstrate this task, we aim to test if a neighbourhood N_s presents a statistically significant accumulation of the therapeutic use for the compounds used to treat the samples in N_s. In order to discover the enriched neighbourhoods, we first define a neighbourhood N_{s_i} for every single sample in the dataset, and thus, samples will generally be included in several neighbourhoods. Then, we attempt to reject the null hypothesis by computing the probability that the property of interest appears in N_{s_i} by chance. Depending on the type of variable, different statistical methods for computing such probability could be required. In this example, illustrated in Figure 8.7, we searched for enrichments related to the therapeutic use of the compounds (i.e. anti-inflammatory, anti-infective, etc) which are binary variables. Each therapeutic use in the database of compounds was evaluated separately, for which the Hypergeometric test was used with a Bonferroni correction to account for the multiple hypothesis testing. Figure 8.7 shows two examples in an ASPP2 screening with Pharmakon 1600 (Figure 8.3, where neighbourhoods were found to be have a statistically significant ($p < 0.01$) presence of compounds with the same therapeutic use: anti-infective and cardiotonic.

8.6 Summary and limitations

We have presented a set of data visualization tools that aim to facilitate the task of exploration of the image data from a high-throughput screening. It allows the 2-D visualization of the entire dataset according to the visual similarities in the samples, with the

additional capability of controlling the level of detail of the visualization (i.e. enhancing local of global structures of the dataset). Moreover, the tools can produce illustrations to transmit to the user the similarities that are being taken into consideration as these can be difficult to recognize even when the clusters are observed. Finally, if additional data about the perturbagens is available, the tool can produce hypotheses by computing enrichments of properties of the perturbagens over every cluster of samples in the dataset.

We note that even though the method can handle high-dimensional descriptors, the effectiveness of the visualizations would increase inversely proportional to the intrinsic dimensionality of the data. The reason is that representation errors appearing with the 2-D projection of the data can be unavoidable, regardless of the dimensionality reduction method, if the data happens to lie in a high-dimensional manifold.

Finally, we note that the pipeline presented in this chapter is a general one that can be applied to experimental setups other than cell imaging with fluorescence microscopy, including other microscopy modalities, organism, or even visual data outside the microscopy domain.

Chapter 9: Summary and future work

To conclude, we summarise the contributions of this thesis and comment on the possible areas for future work. This chapter is organized according to the three main areas of research covered throughout Chapters 3-8.

9.1 Detecting objects in microscopy images

The main contribution for this area within the thesis is a method for object detection in microscopy images which is particularly suitable for images with multiple overlapping instances of an object, and that was presented in Chapter 3 and Chapter 4. Depending on the difficulty of the detection task, the model has the flexibility to choose to detect overlapping objects in groups containing a variable number of instances, as well as individual instances if the task is easy. Such ability to pick the optimal level of granularity is seamlessly obtained during the learning of the model. The inference in the model is computationally efficient, requiring only a few hundred classifier evaluations followed by tree-based dynamic programming. The learning of the model is driven by the proposed instance-count loss (Section 4.3), and requires only simple dot-annotations.

To handle particularly challenging scenarios such as detection on noisy microscopy imaging modalities, we introduced in Chapter 5 a pre-processing module which takes the input images and generates a smooth and contrast-enhanced surface that is optimized for the collection of extremal regions as object detection candidates. We found this generated surface to be helpful in most of our experiments with overlapping instances, not helpful in the cases of mostly non-overlapping instances, and harmful in the case of the synthetic

dataset which contains large clusters of extremely overlapping instances. Variants of the surface could be produced in different ways that could be more appropriate for cases where the objects of interest have a much more complex appearance such as in human detection. One example of an alternative surface would be to compute a pixel-wise probability map of individual object detections.

The proposed detection method is suitable for processing batches of data, for example, coming from high-throughput screenings. In such a scenario, time is not normally a critical constraint, and it is therefore feasible to use a method based on supervised learning, which requires data annotation and model training. Moreover, for use cases where the experimental setup is standard, the annotation and training efforts are only required once, making the system more practical.

Even though extremal regions were chosen for the generation of tree of candidate regions, we recall that any method that produces nested candidate regions such that they result in tree-structured graphical models can make direct use of the learning method and inference procedure. For example, recursive spectral clustering or superpixel merging. However, we found that the quality of the pool of candidate regions is a key issue as good delineation of the objects of interest seems to facilitate learning good features for the classification stage.

We also note that the pool of candidates is not limited to 2D regions as the nestedness condition can be preserved in 3D regions (i.e. 3D MSERs [49]), which could allow a straightforward extension of the method for 3D data. Arguably, the main conceptual difficulty for such extension is the hardness of obtaining dotted annotations for 3D images. Nevertheless, the extension of the method for detection of objects in 3D microscopy is a possible next direction of research.

9.2 Estimating object densities in microscopy images

In Chapter 6, one key contribution towards object density estimation and object counting is the simple ridge-regression framework (Section 6.2) to learn the mapping between

local image features and object density, which was shown to accelerate the learning of the mapping while maintaining the accuracy of the reference method of Lempitsky and Zisserman [90]. Additionally, we proposed a general extension to density estimation methods in order to improve the temporal smoothness (Section 6.3) of the density maps for the cases of counting on time-lapse sequences, and its use was demonstrated (Section 6.4.7) for the ridge-regression framework as well as [90].

With the capability to learn the mapping from visual features to object density on the fly, we proposed in Chapter 7 and interactive counting system. This system was a first foray into enabling counting, previously treated as a traditional batch learning problem, to be handled interactively. To do this we proposed solutions to two additional problems of an interactive counting system. First, we showed a method to deal with the problem of over- or under-fitting of the visual vocabulary through a visual dictionary that grows as the user provides annotations (Section 7.2). Secondly, we proposed two different techniques for the visualization of the object density map in order to guide the user (Section 7.3), in an intuitive manner, towards providing further annotations in the image regions where they are most needed. The result is an agile and flexible system which enables quite disparate visual material (spanning both microscopy images of cells and satellite imagery) to be annotated and counted in a matter of seconds.

One key area of improvement for object density estimation is in the generation of better features that can allow more complex objects to be parsed, and a general architecture for this problem can arise from deep learning, as discussed in Section 9.4 below.

9.3 Exploring microscopy datasets

In Chapter 8, we presented a set of tools that aim to facilitate the task of exploration of the image data from a high-throughput screening. Firstly, we showed how t-SNE [155] in combination with diffusion processes [50] can be used in order to generate 2-D visualization of an entire dataset according to the visual similarities in the samples ((Section 8.3), with the additional capability of controlling the level of detail of the

visualization (i.e. enhancing local of global structures of the dataset). Secondly, we showed how we could take advantage of the Fisher Vector encoding [121] to produce saliency maps which, given a group of visually similar samples, we could use to highlight the visual patterns common in the images of those samples (Section 8.4). Finally, we showed how to use the dataset pairwise similarities determined from the visual information in order to generate quantitative hypotheses of the possible factors (i.e. from additional data about the perturbagens) involved in the generation of the visual patterns of interest (Section 8.5).

So far, this work relies on the encoding power of Fisher vectors on top of dense SIFT features in order to determine the similarities between samples. Nevertheless, such pipeline has been outperformed by modern deep architectures. Therefore, using deep features learned from large microscopy datasets is also promising research area, as further commented in Section 9.4 below.

9.4 Further future work

With the recent success of the so called deep learning architectures in the computer vision community, it has become clear that learning image features from large datasets can improve the accuracy over hand-crafted features for tasks such as image classification [31] and object detection [62]. Therefore, deep architectures are also gaining popularity in areas very related to the work in this thesis such as object counting [134, 171], or object detection in microscopy images [42]. We believe that deep learning can contribute within the field of microscopy image analysis in the same way that it is advancing general computer vision. For instance, learning better features for the relevant tasks such as counting or detection, as well as unsupervised feature learning for the task of whole-image characterization, is a clear area of improvement and very promising research direction. Furthermore, feature learning is just one aspect amongst all the possibilities. Therefore, we have begun the exploration of deep learning architectures for the tasks of detection and counting of overlapping instances of an objects in natural images, with the hope that

the lessons learned and methods developed can then contribute towards advancing the field of microscopy image analysis.

Bibliography

[1] Y. Al-Kofahi, W. Lassoued, W. Lee, and B. Roysam. Improved automatic detection and segmentation of cell nuclei in histopathology images. *IEEE Transactions on Biomedical Engineering*, 57(4):841 –852, Apr 2010.

[2] R. Ali, M. Gooding, T. Szilágyi, B. Vojnovic, M. Christlieb, and M. Brady. Automatic segmentation of adherent biological cell boundaries and nuclei from brightfield microscopy images. *Machine Vision and Applications*, 23(4):607–621, 2012.

[3] C. Allan, J.-M. Burel, J. Moore, C. Blackburn, M. Linkert, S. Loynton, D. MacDonald, W. J. Moore, C. Neves, A. Patterson, et al. Omero: flexible, model-driven data management for experimental biology. *Nature methods*, 9(3):245–253, 2012.

[4] E.-a. D. Amir, K. L. Davis, M. D. Tadmor, E. F. Simonds, J. H. Levine, S. C. Bendall, D. K. Shenfeld, S. Krishnaswamy, G. P. Nolan, and D. Pe'er. visne enables visualization of high dimensional single-cell data and reveals phenotypic heterogeneity of leukemia. *Nature biotechnology*, 31(6):545–552, 2013.

[5] R. Arandjelović and A. Zisserman. Three things everyone should know to improve object retrieval. In *Proc. CVPR*, 2012.

[6] P. Arbelaez, M. Maire, C. Fowlkes, and J. Malik. Contour detection and hierarchical image segmentation. *TPAMI*, 33:898–916, 2011.

[7] J. Arsenio, B. Kakaradov, P. J. Metz, S. H. Kim, G. W. Yeo, and J. T. Chang. Early specification of cd8+ t lymphocyte fates during adaptive immunity revealed by single-cell gene-expression analyses. *Nature immunology*, 15(4):365–372, 2014.

[8] C. Arteta, V. Lempitsky, J. A. Noble, and A. Zisserman. Learning to count objects in images: project webpage. `http://www.robots.ox.ac.uk/~vgg/research/counting/`.

[9] C. Arteta, V. Lempitsky, J. A. Noble, and A. Zisserman. Learning to detect cells using non-overlapping extremal regions. In N. Ayache, editor, *International Conference on Med-*

ical Image Computing and Computer Assisted Intervention, Lecture Notes in Computer Science, pages 348–356. MICCAI, Springer, 2012.

[10] C. Arteta, V. Lempitsky, J. A. Noble, and A. Zisserman. Learning to detect partially overlapping instances. In *Proc. CVPR*, 2013.

[11] C. Arteta, V. Lempitsky, J. A. Noble, and A. Zisserman. Interactive object counting. In *European Conference on Computer Vision*, 2014.

[12] C. Arteta, V. Lempitsky, J. A. Noble, and A. Zisserman. Detecting overlapping instances in microscopy images using extremal region trees. *Medical image analysis*, 2015.

[13] J. J. Babcock and M. Li. Deorphanizing the human transmembrane genome: A landscape of uncharacterized membrane proteins. *Acta pharmacologica Sinica*, 35(1):11–23, 2014.

[14] A. Bahnson, C. Athanassiou, D. Koebler, L. Qian, T. Shun, D. Shields, H. Yu, H. Wang, J. Goff, T. Cheng, R. Houck, and L. Cowsert. Automated measurement of cell motility and proliferation. *BMC Cell Biology*, 6(1):19, 2005.

[15] O. Barinova, V. Lempitsky, and P. Kholi. On detection of multiple object instances using hough transforms. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 34(9):1773–1784, 2012.

[16] C. Bath, S. Yang, D. Muttuvelu, T. Fink, J. Emmersen, H. Vorum, J. Hjortdal, and V. Zachar. Hypoxia is a key regulator of limbal epithelial stem cell growth and differentiation. *Stem cell research*, 10(3):349–360, 2013.

[17] B. Becher, A. Schlitzer, J. Chen, F. Mair, H. R. Sumatoh, K. W. W. Teng, D. Low, C. Ruedl, P. Riccardi-Castagnoli, M. Poidinger, et al. High-dimensional analysis of the murine myeloid cell system. *Nature immunology*, 15(12):1181–1189, 2014.

[18] S. Belongie and J. Malik. Shape matching and object recognition using shape contexts. *TPAMI*, 24(24), 2002.

[19] A. B. Berger, G. G. Cabal, E. Fabre, T. Duong, H. Buc, U. Nehrbass, J.-C. Olivo-Marin, O. Gadal, and C. Zimmer. High-resolution statistical mapping reveals gene territories in live yeast. *Nature Methods*, 5(12):1031–1037, 2008.

[20] E. Bernardis and S. X. Yu. Pop out many small structures from a very large microscopic image. *Med. Image Analysis*, 15(5):690–707, 2011.

[21] C. Bishop. *Pattern Recognition and Machine Learning*. Springer, New York, 2006.

[22] G. B. Blanchard, A. J. Kabla, N. L. Schultz, L. C. Butler, B. Sanson, N. Gorfinkiel, L. Mahadevan, and R. J. Adams. Tissue tectonics: morphogenetic strain rates, cell shape change and intercalation. *Nature Methods*, 6(6):458–464, 2009.

[23] R. Borgo, J. Kehrer, D. H. Chung, E. Maguire, R. S. Laramee, H. Hauser, M. Ward, and M. Chen. Glyph-based visualization: Foundations, design guidelines, techniques and applications. *Eurographics State of the Art Reports*, pages 39–63, 2013.

[24] Y. Boykov and M. P. Jolly. Interactive graph cuts for optimal boundary and region segmentation of objects in N-D images. In *Proc. ICCV*, volume 2, pages 105–112, 2001.

[25] C. Brechbühler, G. Gerig, and O. Kübler. Parametrization of closed surfaces for 3-d shape description. *Computer Vision and Image Understanding*, 61(2):154–170, Mar 1995.

[26] L. Breiman. Random forests. 45(1):5–32, 2001.

[27] A. E. Carpenter, T. R. Jones, M. R. Lamprecht, C. Clarke, I. H. Kang, O. Friman, D. A. Guertin, J. H. Chang, R. A. Lindquist, J. Moffat, P. Golland, and D. M. Sabatini. CellProfiler: image analysis software for identifying and quantifying cell phenotypes. *Genome Biology*, 7(10):R100, Oct 2006.

[28] A. E. Carpenter, L. Kamentsky, and K. W. Eliceiri. A call for bioimaging software usability. *Nature methods*, 9(7):666–670, 2012.

[29] A. B. Chan, Z.-S. J. Liang, and N. Vasconcelos. Privacy preserving crowd monitoring: Counting people without people models or tracking. In *Proc. CVPR*, 2008.

[30] K. Chatfield, V. Lempitsky, A. Vedaldi, and A. Zisserman. The devil is in the details: an evaluation of recent feature encoding methods. In *Proc. BMVC.*, 2011.

[31] K. Chatfield, K. Simonyan, A. Vedaldi, and A. Zisserman. Return of the devil in the details: Delving deep into convolutional nets. In *British Machine Vision Conference*, 2014.

[32] K. Chatfield, A. Vedaldi, L. Victor, and A. Zisserman. The devil is in the details: an evaluation of recent feature encoding methods webpage. `http://www.robots.ox.ac.uk/~vgg/research/encoding_eval`.

[33] Q. Chaudry, S. Raza, A. Young, and M. Wang. Automated renal cell carcinoma subtype classification using morphological, textural and wavelets based features. *Journal of Signal Processing Systems*, 55(1):15–23, 2009.

[34] S.-C. Chen, G. J. Gordon, and R. F. Murphy. Graphical models for structured classification, with an application to interpreting images of protein subcellular location patterns. *J. Mach. Learn. Res.*, 9:651–682, Jun 2008.

[35] T. Chen, Y. Zhang, C. Wang, Z. Qu, M. Cai, F. Wang, and T. Syeda-Mahmood. Local complex phase based level set and its application to DIC red blood cell segmentation. In *2011 IEEE International Symposium on Biomedical Imaging: From Nano to Macro*, pages 187 –190, Apr 2011.

[36] X. Chen, R. F. Murphy, et al. Objective clustering of proteins based on subcellular location patterns. *Journal of Biomedicine and Biotechnology*, 2:87, 2005.

[37] X. Chen, M. Velliste, and R. F. Murphy. Automated interpretation of subcellular patterns in fluorescence microscope images for location proteomics. *Cytometry Part A*, 69(7):631–640, 2006.

[38] J. Cheng, M. Veronika, and J. Rajapakse. Identifying cells in histopathological images. *Recognizing Patterns in Signals, Speech, Images and Videos*, page 244–252, 2010.

[39] J. Cheng, M. Veronika, and J. Rajapakse. Identifying cells in histopathological images. *In proc. ICPR Contest*, pages 244–252, 2010.

[40] A. B. Chetverin and H. V. Chetverina. Method for amplification of nucleic acids in solid media, Apr 1 1997. US Patent 5,616,478.

[41] S.-Y. Cho, T. W. S. Chow, and C.-T. Leung. A neural-based crowd estimation by hybrid global learning algorithm. *Systems, Man, and Cybernetics, Part B: Cybernetics, IEEE Transactions on*, 29(4):535–541, Aug 1999.

[42] D. C. Cireşan, A. Giusti, L. M. Gambardella, and J. Schmidhuber. Mitosis detection in breast cancer histology images with deep neural networks. In *Medical Image Computing and Computer-Assisted Intervention–MICCAI 2013*, pages 411–418. Springer, 2013.

[43] C. Cortes and V. Vapnik. Support-vector networks. *Machine Learning*, 20(3):273–297, 1995.

[44] R. Creton. Automated analysis of behavior in zebrafish larvae. *Behavioural brain research*, 203(1):127–136, 2009.

[45] N. Dalal and B. Triggs. Histograms of oriented gradients for human detection. In *Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference on*, volume 1, page 886–893, 2005.

[46] O. Debeir, I. Adanja, N. Warzee, P. Van Ham, and C. Decaestecker. Phase contrast image segmentation by weak watershed transform assembly. In *5th IEEE International Symposium on Biomedical Imaging: From Nano to Macro, 2008. ISBI 2008*, pages 724 –727, May 2008.

[47] I. S. Dhillon, Y. Guan, and B. Kulis. Weighted graph cuts without eigenvectors a multilevel approach. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 29(11):1944–1957, 2007.

[48] L. Dong, V. Parameswaran, V. Ramesh, and I. Zoghlami. Fast crowd segmentation using shape indexing. In *Proc. ICCV*, 2007.

[49] M. Donoser and H. Bischof. segmentation by maximally stable volumes (msvs). In *Pattern Recognition, 2006. ICPR 2006. 18th International Conference on*, volume 1, pages 63–66. IEEE, 2006.

[50] M. Donoser and H. Bischof. Diffusion processes for retrieval revisited. In *Computer Vision and Pattern Recognition (CVPR), 2013 IEEE Conference on*, pages 1320–1327. IEEE, 2013.

[51] A. Dufour, V. Shinin, S. Tajbakhsh, N. Guillen-Aghion, J. C. Olivo-Marin, and C. Zimmer. Segmenting and tracking fluorescent cells in dynamic 3-d microscopy with coupled active surfaces. *IEEE Transactions on Image Processing*, 14(9):1396–1410, Sep 2005.

[52] M. Felsberg and G. Sommer. The monogenic signal. *IEEE Transactions on Signal Processing*, 49(12):3136 –3144, Dec 2001.

[53] L. Fiaschi, R. Nair, U. Köethe, and F. Hamprecht. Learning to count with regression forest and structured labels. In *Proc. ICPR*, 2012.

[54] M. A. Fischler and R. C. Bolles. Random sample consensus: A paradigm for model fitting with applications to image analysis and automated cartography. *Comm. ACM*, 24(6):381–395, 1981.

[55] R. A. Fisher. The use of multiple measurements in taxonomic problems. *Annals of Human Genetics*, 7(2):179–188, 1936.

[56] G. Flaccavento, V. Lempitsky, I. Pope, P. Barber, A. Zisserman, and A. Noble. Learning to count cells: Applications to lens-free imaging of large fields. *Conference in Microscopic Image Analysis with Applications in Biology*, Sep 2011.

[57] G. Flaccavento, V. Lempitsky, I. Pope, P. R. Barber, A. Zisserman, J. A. Noble, and B. Vojnovic. Learning to count cells: applications to lens-free imaging of large fields. In *Microscopic Image Analysis with Applications in Biology*, 2011.

[58] N. A. P. Franken, H. M. Rodermond, J. Stap, J. Haveman, and C. van Bree. Clonogenic assay of cells in vitro. *Nat. Protocols*, 1(5):2315–2319, Dec 2006.

[59] J. Friedman. Additive logistic regression: a statistical view of boosting (With discussion and a rejoinder by the authors). *The Annals of Statistics*, 28(2):337–407, Apr 2000. Mathematical Reviews number (MathSciNet): MR1790002; Zentralblatt MATH identifier: 01828945.

[60] A. Gelas, K. Mosaliganti, A. Gouaillard, L. Souhait, R. Noche, N. Obholzer, and S. G. Megason. Variational level-set with gaussian shape model for cell segmentation. In *2009 16th IEEE International Conference on Image Processing (ICIP)*, pages 1089–1092. IEEE, Nov 2009.

[61] M. C. Gibson, A. B. Patel, R. Nagpal, and N. Perrimon. The emergence of geometric order in proliferating metazoan epithelia. *Nature*, 442(7106):1038–1041, 2006.

[62] R. B. Girshick, J. Donahue, T. Darrell, and J. Malik. Rich feature hierarchies for accurate object detection and semantic segmentation. In *Proc. CVPR*, 2014.

[63] F. Graf, M. Grzegorzek, and D. Paulus. Counting lymphocytes in histopathology images using connected components. *In proc. ICPR Contest*, pages 263–269, 2010.

[64] U. Grenander and M. I. Miller. Computational anatomy: An emerging discipline. *Quarterly of applied mathematics*, 56(4):617–694, 1998.

[65] A. Gupta, A. Kembhavi, and L. S. Davis. Observing human-object interactions: Using spatial and functional compatibility for recognition. *TPAMI*, 2009.

[66] M. N. Gurcan, A. Madabhushi, and N. Rajpoot. Pattern recognition in histopathological images: An icpr 2010 contest. In *Recognizing Patterns in Signals, Speech, Images and Videos*, pages 226–234. Springer, 2010.

[67] R. I. Hartley and A. Zisserman. *Multiple View Geometry in Computer Vision*. Cambridge University Press, ISBN: 0521540518, second edition, 2004.

[68] G. E. Hinton and S. T. Roweis. Stochastic neighbor embedding. In *Advances in neural information processing systems*, pages 833–840, 2002.

[69] P. Horvath, T. Wild, U. Kutay, and G. Csucs. Machine learning improves the precision and robustness of high-content screens using nonlinear multiparametric methods to analyze screening results. *Journal of biomolecular screening*, 16(9):1059–1067, 2011.

[70] H. Hotelling. Analysis of a complex of statistical variables into principal components. *Journal of educational psychology*, 24(6):417, 1933.

[71] K. Huang and R. F. Murphy. Boosting accuracy of automated classification of fluorescence microscope images for location proteomics. *BMC Bioinformatics*, 5(1):78, Jun 2004.

[72] H. Idrees, I. Saleemi, C. Seibert, and M. Shah. Multi-source multi-scale counting in extremely dense crowd images. In *Computer Vision and Pattern Recognition (CVPR), 2013 IEEE Conference on*, pages 2547–2554. IEEE, 2013.

[73] S. Isikman, I. Sencan, O. Mudanyali, W. Bishara, C. Oztoprak, and A. Ozcan. Color and monochrome lensless on-chip imaging of caenorhabditis elegans over a wide field-of-view. *Lab on a Chip*, 10(9):1109–1112, 2010.

[74] H. Jégou, M. Douze, C. Schmid, and P. Pérez. Aggregating local descriptors into a compact image representation. In *Proc. CVPR*, 2010.

[75] T. R. Jones, A. E. Carpenter, M. R. Lamprecht, J. Moffat, S. J. Silver, J. K. Grenier, A. B. Castoreno, U. S. Eggert, D. E. Root, P. Golland, and D. M. Sabatini. Scoring diverse cellular morphologies in image-based screens with iterative feedback and machine learning. *Proceedings of the National Academy of Sciences*, 106(6):1826–1831, Oct 2009.

[76] T. R. Jones, I. H. Kang, D. B. Wheeler, R. A. Lindquist, A. Papallo, D. M. Sabatini, P. Golland, and A. E. Carpenter. Cellprofiler analyst: data exploration and analysis software for complex image-based screens. *BMC bioinformatics*, 9(1):482, 2008.

[77] J. D. Kangas, A. W. Naik, and R. F. Murphy. Efficient discovery of responses of proteins to compounds using active learning. *BMC bioinformatics*, 15(1):143, 2014.

[78] M. Kass, A. Witkin, and D. Terzopoulos. Snakes: Active contour models. *International journal of computer vision*, 1(4):321–331, 1988.

[79] K. Khairy, E. Reynaud, and E. Stelzer. Detection of deformable objects in 3D images using markov-chain monte carlo and spherical harmonics. *Medical image computing and computer-assisted intervention: MICCAI ... International Conference on Medical Image Computing and Computer-Assisted Intervention*, 11(Pt 2):1075–1082, 2008. PMID: 18982711.

[80] D. Kong, D. Gray, and H. Tao. A viewpoint invariant approach for crowd counting. In *Proc. ICPR*, volume 3, pages 1187–1190. IEEE, 2006.

[81] H. Kong, M. Gurcan, and K. Belkacem-Boussaid. Partitioning histopathological images: an integrated framework for supervised color-texture segmentation and cell splitting. *IEEE transactions on medical imaging*, 30(9):1661–1677, Sep 2011. PMID: 21486712.

[82] A. Krause, J. Stoye, and M. Vingron. Large scale hierarchical clustering of protein sequences. *BMC bioinformatics*, 6(1):15, 2005.

[83] M. Kuse, T. Sharma, and S. Gupta. A classification scheme for lymphocyte segmentation in H&E stained histology images. *Recognizing Patterns in Signals, Speech, Images and Videos*, page 235?243, 2010.

[84] M. Kuse, Y. Wang, V. Kalasannavar, M. Khan, N. Rajpoot, et al. Local isotropic phase symmetry measure for detection of beta cells and lymphocytes. *Journal of Pathology Informatics*, 2(2):2, 2011.

[85] T. Lecuit and P.-F. Lenne. Cell surface mechanics and the control of cell shape, tissue patterns and morphogenesis. *Nature Reviews Molecular Cell Biology*, 8(8):633–644, 2007.

[86] Y. LeCun. The MNIST database of handwritten digits. http://yann.lecun.com/exdb/mnist/.

[87] K.-M. Lee and W. N. Street. Model-based detection, segmentation, and classification for image analysis using on-line shape learning. *Machine Vision and Applications*, 13(4):222–233, 2003.

[88] A. Lehmussola, P. Ruusuvuori, J. Selinummi, H. Huttunen, and O. Yli-Harja. Computational framework for simulating fluorescence microscope images with cell populations. *Medical Imaging, IEEE Transactions on*, 26(7):1010–1016, 2007.

[89] V. Lempitsky, A. Vedaldi, and A. Zisserman. A pylon model for semantic segmentation. In *NIPS*, 2011.

[90] V. Lempitsky and A. Zisserman. Learning to count objects in images. In *NIPS*, 2010.

[91] G. Li, T. Liu, A. Tarokh, J. Nie, L. Guo, A. Mara, S. Holley, and S. T. Wong. 3D cell nuclei segmentation based on gradient flow tracking. *BMC Cell Biology*, 8(1):40, Sep 2007.

[92] P. Liberali, B. Snijder, and L. Pelkmans. Single-cell and multivariate approaches in genetic perturbation screens. *Nature Reviews Genetics*, 16(1):18–32, 2015.

[93] G. Lin, M. K. Chawla, K. Olson, J. F. Guzowski, C. A. Barnes, and B. Roysam. Hierarchical, model-based merging of multiple fragments for improved three-dimensional segmentation of nuclei. *Cytometry. Part A: the journal of the International Society for Analytical Cytology*, 63(1):20–33, 2005. PMID: 15584021.

[94] K. Logg, A. Diez, K. Bodvard, M. Kńll, et al. Image analysis algorithms for cell contour recognition in budding yeast. *Optics express*, 16(17):12943–12957, 2008.

[95] X. Lou and F. Hamprecht. Structured learning from partial annotations. *arXiv:1206.6421*, Jun 2012.

[96] D. Lowe. Distinctive image features from scale-invariant keypoints. *IJCV*, 60(2):91–110, 2004.

[97] W. Ma, L. Huang, and C. Liu. Crowd density analysis using co-occurrence texture features. In *Computer Sciences and Convergence Information Technology (ICCIT), 2010 5th International Conference on*, pages 170–175, Nov 2010.

[98] N. Malpica, C. Ortiz de Solorzano, J. J. Vaquero, A. Santos, I. Vallcorba, J. M. Garcia-Sagredo, and F. del Pozo. Applying watershed algorithms to the segmentation of clustered nuclei. *Cytometry*, 28(4):289–297, 1997.

[99] N. Malpica, A. Santos, A. Tejedor, A. Torres, M. Castilla, P. García-Barreno, and M. Desco. Automatic quantification of viability in epithelial cell cultures by texture analysis. *Journal of Microscopy*, 209(1):34–40, 2003.

[100] K. Mao, P. Zhao, and P.-H. Tan. Supervised learning-based cell image segmentation for p53 immunohistochemistry. *IEEE Transactions on Biomedical Engineering*, 53(6):1153 –1163, Jun 2006.

[101] A. Marana, S. Velastin, L. Costa, and R. Lotufo. Estimation of crowd density using image processing. In *Image Processing for Security Applications, IEE Colloquium on*, pages 11–1. IET, 1997.

[102] M. Marcuzzo, P. Quelhas, A. Campilho, A. M. Mendonça, and A. Campilho. Automated arabidopsis plant root cell segmentation based on SVM classification and region merging. *Computers in biology and medicine*, 39(9):785–793, Sep 2009. PMID: 19604506.

[103] G. Martinez, J. G. Frerichs, K. Joeris, K. Konstantinov, and T. Scheper. Cell density estimation from a still image for in-situ microscopy. In *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing*, page 18–23, 2005.

[104] J. Matas, O. Chum, M. Urban, and T. Pajdla. Robust wide-baseline stereo from maximally stable extremal regions. *Image and vision computing*, 22(10):761–767, 2004.

[105] C. Mayer, S. Dimopoulos, F. Rudolf, and J. Stelling. Using cellx to quantify intracellular events. *Current Protocols in Molecular Biology*, pages 14–22, 2013.

[106] T. McInerney and D. Terzopoulos. T-snakes: Topology adaptive snakes. *Medical Image Analysis*, 4(2):73–91, Jun 2000.

[107] R. D. Mitra and G. M. Church. In situ localized amplification and contact replication of many individual dna molecules. *Nucleic Acids Research*, 27(24):e34–e39, 1999.

[108] D. P. Mukherjee, N. Ray, and S. T. Acton. Level set analysis for leukocyte detection and tracking. *IEEE Transactions on Image Processing*, 13(4):562–572, Apr 2004.

[109] P. Mundra and J. Rajapakse. SVM-RFE with MRMR filter for gene selection. *IEEE Transactions on NanoBioscience*, 9(1):31 –37, Mar 2010.

[110] R. F. Murphy. An active role for machine learning in drug development. *Nature Chemical Biology*, 7(6):327–330, 2011.

[111] S. Nath, K. Palaniappan, and F. Bunyak. Cell segmentation using coupled level sets and graph-vertex coloring. *Medical Image Computing and Computer-Assisted Intervention–MICCAI 2006*, page 101–108, 2006.

[112] T. W. Nattkemper, H. Wersing, W. Schubert, and H. Ritter. A neural network architecture for automatic segmentation of fluorescence micrographs. *Neurocomputing*, 48(1–4):357–367, Oct 2002.

[113] J. Newberg, J. Li, A. Rao, F. Ponten, M. Uhlen, E. Lundberg, and R. Murphy. Automated analysis of human protein atlas immunofluorescence images. In *IEEE International Symposium on Biomedical Imaging: From Nano to Macro, 2009. ISBI '09*, pages 1023 –1026, Jul 2009.

[114] I. Okoli, J. J. Coleman, E. Tampakakis, W. F. An, E. Holson, F. Wagner, A. L. Conery, J. Larkins-Ford, G. Wu, A. Stern, et al. Identification of antifungal compounds active against candida albicans using an improved high-throughput caenorhabditis elegans assay. *PLoS One*, 4(9):e7025, 2009.

[115] J. Pan, T. Kanade, and M. Chen. Learning to detect different types of cells under phase contrast microscopy. *Microscopic Image Analysis with Applications in Biology (MIAAB)*, 2009.

[116] C. Panagiotakis, E. Ramasso, and G. Tziritas. Lymphocyte segmentation using the transferable belief model. *Recognizing Patterns in Signals, Speech, Images and Videos*, page 253–262, 2010.

[117] C. Panagiotakis, E. Ramasso, and G. Tziritas. Lymphocyte segmentation using the transferable belief model. *In proc. ICPR Contest*, pages 253–262, 2010.

[118] J. Pearl. *Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference*. Morgan Kauffman, San Mateo, California, 1988.

[119] H. Peng, X. Zhou, F. Li, X. Xia, and S. T. C. Wong. Integrating multi-scale blob/curvilinear detector techniques and multilevel sets for automated segmentation of stem cell images. In *Proceedings of the Sixth IEEE international conference on Symposium on Biomedical Imaging: From Nano to Macro*, ISBI'09, page 1362–1365, Piscataway, NJ, USA, 2009. IEEE Press.

[120] T. Peng and R. F. Murphy. Image-derived, three-dimensional generative models of cellular organization. *Cytometry Part A*, 79A(5):383–391, 2011.

[121] F. Perronnin, J. Sánchez, and T. Mensink. Improving the Fisher kernel for large-scale image classification. In *Proc. ECCV*, 2010.

[122] Z. Pincus and J. A. Theriot. Comparison of quantitative methods for cell-shape analysis. *Journal of Microscopy*, 227(2):140–156, 2007.

[123] C. E. Rasmussen. Gaussian processes for machine learning. 2006.

[124] J. Rittscher. Characterization of biological processes through automated image analysis. *Annual review of biomedical engineering*, 12:315–344, Aug 2010. PMID: 20482277.

[125] N. Robertson, D. Sanders, P. Seymour, and R. Thomas. The four-colour theorem. *journal of combinatorial theory, Series B*, 70(1):2–44, 1997.

[126] G. Rohde, W. Wang, T. Peng, and R. Murphy. Deformation-based nonlinear dimension reduction: Applications to nuclear morphometry. In *5th IEEE International Symposium on Biomedical Imaging: From Nano to Macro, 2008. ISBI 2008*, pages 500 –503, May 2008.

[127] S. T. Roweis and L. K. Saul. Nonlinear dimensionality reduction by locally linear embedding. *Science*, 290:2323–2326, 2000.

[128] D. Ryan, S. Denman, C. Fookes, and S. Sridharan. Crowd counting using multiple local features. In *Proc. DICTA*, pages 81–88. IEEE, 2009.

[129] H. Z. Sailem, J. E. Sero, and C. Bakal. Visualizing cellular imaging data using phenoplot. *Nature communications*, 6, 2015.

[130] T. R. Samatov, H. V. Chetverina, and A. B. Chetverin. Real-time monitoring of dna colonies growing in a polyacrylamide gel. *Analytical biochemistry*, 356(2):300–302, 2006.

[131] J. W. Sammon. A nonlinear mapping for data structure analysis. *IEEE Transactions on computers*, (5):401–409, 1969.

[132] A. Sarti, R. Malladi, and J. Sethian. Subjective surfaces: A geometric model for boundary completion. *International Journal of Computer Vision*, 46(3):201–221, 2002.

[133] J. Schindelin, I. Arganda-Carreras, E. Frise, V. Kaynig, M. Longair, T. Pietzsch, S. Preibisch, C. Rueden, S. Saalfeld, B. Schmid, et al. Fiji: an open-source platform for biological-image analysis. *Nature methods*, 9(7):676–682, 2012.

[134] S. Segui, O. Pujol, and J. Vitria. Learning to count with deep object features. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, pages 90–96, 2015.

[135] E.-Y. Seo, T.-S. Ahn, and Y.-G. Zo. Agreement, precision, and accuracy of epifluorescence microscopy methods for enumeration of total bacterial numbers. *Applied and Environmental Microbiology*, 76(6):1981–1991, Mar 2010.

[136] J. A. Sethian et al. Level set methods and fast marching methods. *Journal of Computing and Information Technology*, 11(1):1–2, 2003.

[137] J. Shan, R. E. Schwartz, N. T. Ross, D. J. Logan, D. Thomas, S. A. Duncan, T. E. North, W. Goessling, A. E. Carpenter, and S. N. Bhatia. Identification of small molecules for human hepatocyte expansion and ips differentiation. *Nature chemical biology*, 9(8):514–520, 2013.

[138] J. Shi and J. Malik. Normalized cuts and image segmentation. *TPAMI*, 22(8):888–905, 2000.

[139] T. Shimada, K. Kato, A. Kamikouchi, and K. Ito. Analysis of the distribution of the brain cells of the fruit fly by an automatic cell counting algorithm. *Physica A: Statistical Mechanics and its Applications*, 350(1):144–149, May 2005.

[140] K. Simonyan, O. M. Parkhi, A. Vedaldi, and A. Zisserman. Fisher Vector Faces in the Wild. In *Proc. BMVC.*, 2013.

[141] S. Singh, F. Janoos, T. Pécot, E. Caserta, K. Huang, J. Rittscher, G. Leone, and R. Machiraju. Non-parametric population analysis of cellular phenotypes. In G. Fichtinger, A. Martel, and T. Peters, editors, *Medical Image Computing and Computer-Assisted Intervention – MICCAI 2011*, volume 6892 of *Lecture Notes in Computer Science*, pages 343–351. Springer Berlin / Heidelberg, 2011.

[142] J. Sivic and A. Zisserman. Video Google: A text retrieval approach to object matching in videos. In *Proc. ICCV*, volume 2, pages 1470–1477, 2003.

[143] P. Skandamis, T. F. Brocklehurst, E. Panagou, and G.-J. Nychas. Image analysis as a mean to model growth of escherichia coli O157:H7 in gel cassettes. *Journal of Applied Microbiology*, 103(4):937–947, 2007.

[144] K. Smith and V. Lepetit. General constraints for batch multiple-target tracking applied to largescale videomicroscopy. In *Proc. CVPR*, 2008.

[145] P. Soille. *Morphological image analysis: principles and applications*. Springer-Verlag New York, Inc., 2003.

[146] C. Sommer, C. Straehle, U. Koethe, F. Hamprecht, et al. ilastik: Interactive learning and segmentation toolkit. In *Biomedical Imaging: From Nano to Macro, 2011 IEEE International Symposium on*, pages 230–233. IEEE, 2011.

[147] G. Srinivasa, M. C. Fickus, Y. Guo, A. D. Linstedt, and J. Kovačević. Active mask segmentation of fluorescence microscope images. *IEEE transactions on image processing : a publication of the IEEE Signal Processing Society*, 18(8):1817–1829, Aug 2009. PMID: 19380268 PMCID: PMC2765110.

[148] M. Szummer, P. Kohli, and D. Hoiem. Learning CRFs using graph cuts. In D. Forsyth, P. Torr, and A. Zisserman, editors, *Computer Vision – ECCV 2008*, volume 5303 of *Lecture Notes in Computer Science*, pages 582–595. Springer Berlin / Heidelberg, 2008.

[149] J. B. Tenenbaum, V. de Silva, and J. C. Langford. A global geometric framework for nonlinear dimensionality reduction. *Science*, 290:2319–2323, 2000.

[150] D. Theriault, M. Walker, J. Wong, and M. Betke. Cell morphology classification and clutter mitigation in phase-contrast microscopy images using machine learning. *Machine Vision and Applications*, 23(4):659–673, 2012.

[151] W. S. Torgerson. Multidimensional scaling: I. theory and method. *Psychometrika*, 17(4):401–419, 1952.

[152] I. Tsochantaridis, T. Hofmann, T. Joachims, and Y. Altun. Support vector machine learning for interdependent and structured output spaces. In *Proc. ICML*, 2004.

[153] S. C. Turaga, K. L. Briggman, M. Helmstaedter, W. Denk, and H. S. Seung. Maximin affinity learning of image segmentation. In *NIPS*, pages 1865–1873, 2009.

[154] M. Turk and A. P. Pentland. Face recognition using eigenfaces. In *Proc. CVPR*, pages 586–591, 1991.

[155] L. Van der Maaten and G. Hinton. Visualizing data using t-SNE. *Journal of Machine Learning Research*, 2008.

[156] V. N. Vapnik. *The Nature of Statistical Learning Theory*. Springer-Verlag, 1999.

[157] M. Varma and A. Zisserman. Texture classification: Are filter banks necessary? In *Proc. CVPR*, volume 2, pages 691–698, 2003.

[158] A. Vedaldi and B. Fulkerson. VLFeat - an open and portable library of computer vision algorithms. In *ACM Multimedia*, 2010.

[159] Y. Verdié and F. Lafarge. Detecting parametric objects in large scenes by monte carlo sampling. *International Journal of Computer Vision*, pages 1–19, 2013.

[160] B. Vojnovic, P. Barber, I. Pope, P. Smith, and R. Errington. Detecting objects, Jan 22 2008. US Patent App. 12/523,878.

[161] C. Wählby, A. L. Conery, M.-A. Bray, L. Kamentsky, J. Larkins-Ford, K. L. Sokolnicki, M. Veneskey, K. Michaels, A. E. Carpenter, and E. J. O'Rourke. High-and low-throughput scoring of fat mass and body fat distribution in c. elegans. *Methods*, 68(3):492–499, 2014.

[162] W. Wang, Y. Mo, J. Ozolek, and G. Rohde. Characterizing morphology differences from image data using a modified fisher criterion. In *2011 IEEE International Symposium on Biomedical Imaging: From Nano to Macro*, pages 129 –132, Apr 2011.

[163] X. Wang, W. He, D. Metaxas, R. Mathew, and E. White. Cell segmentation and tracking using texture-adaptive snakes. In *4th IEEE International Symposium on Biomedical Imaging: From Nano to Macro, 2007. ISBI 2007*, pages 101–104. IEEE, Apr 2007.

[164] N. Wei, J. You, K. Friehs, E. Flaschel, and T. W. Nattkemper. An in situ probe for on[U+2010]line monitoring of cell density and viability on the basis of dark field microscopy in conjunction with image processing and supervised machine learning. *Biotechnology and Bioengineering*, 97(6):1489–1500, Aug 2007.

[165] S. Wienert, D. Heim, K. Saeger, A. Stenzinger, M. Beil, P. Hufnagl, M. Dietel, C. Denkert, and F. Klauschen. Detection and segmentation of cell nuclei in virtual microscopy images: a minimum-model approach. *Scientific reports*, 2:503, 2012. PMID: 22787560.

[166] X. Wu and S. Shah. A bottom-up and top-down model for cell segmentation using multispectral data. In *2010 IEEE International Symposium on Biomedical Imaging: From Nano to Macro*, pages 592 –595, Apr 2010.

[167] C. Xu and J. Prince. Snakes, shapes, and gradient vector flow. *IEEE Transactions on Image Processing*, 7(3):359 –369, Mar 1998.

[168] Z. Yin, R. Bise, M. Chen, and T. Kanade. Cell segmentation in microscopy imagery using a bag of local bayesian classifiers. In *2010 IEEE International Symposium on Biomedical Imaging: From Nano to Macro*, pages 125–128. IEEE, Apr 2010.

[169] C. Yu and T. Joachims. Learning structural svms with latent variables. In *Proc. ICML*, pages 1169–1176, New York, NY, USA, 2009. ACM.

[170] C. Zhang, J. Yarkony, and F. Hamprecht. Cell detection and segmentation using correlation clustering. In P. Golland, N. Hata, C. Barillot, J. Hornegger, and R. Howe, editors, *Medical Image Computing and Computer-Assisted Intervention – MICCAI 2014*, volume 8673 of *Lecture Notes in Computer Science*, pages 9–16. Springer International Publishing, 2014.

[171] J. Zhang, M. Sameki, S. Ma, B. Price, R. Mech, X. Shen, M. Betke, S. Sclaroff, and Z. Lin. Salient object subitizing. In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2015.

[172] T. Zhao and R. F. Murphy. Automated learning of generative models for subcellular location: Building blocks for systems biology. *Cytometry Part A*, 71A(12):978–990, 2007.

[173] T. Zhao, M. Velliste, M. Boland, and R. Murphy. Object type recognition for automated analysis of protein subcellular location. *IEEE Transactions on Image Processing*, 14(9):1351 –1359, Sep 2005.

[174] G. Zheng, S. Lee, Y. Antebi, M. Elowitz, and C. Yang. The epetri dish, an on-chip cell imaging platform based on subpixel perspective sweeping microscopy (spsm). *Proceedings of the National Academy of Sciences*, 108(41):16889–16894, 2011.

[175] C. Zimmer, E. Labruyere, V. Meas-Yedid, N. Guillen, and J. C. Olivo-Marin. Segmentation and tracking of migrating cells in videomicroscopy with parametric active contours: a tool

for cell-based drug testing. *Medical Imaging, IEEE Transactions on*, 21(10):1212–1221, 2002.