Detecting Overlapping Instances in Microscopy Images Using Extremal Region Trees

Carlos Arteta[a,*], Victor Lempitsky[b], J. Alison Noble[a], Andrew Zisserman[a]

[a]*Department of Engineering Science, University of Oxford, UK*
[b]*Skolkovo Institute of Science and Technology (Skoltech), Russia*

Abstract

In many microscopy applications the images may contain both regions of low and high cell density corresponding to different tissues or colonies at different stages of growth. This poses a challenge to most previously developed automated cell detection and counting methods, which are designed to handle either the low-density scenario (through cell detection) or the high-density scenario (through density estimation or texture analysis).

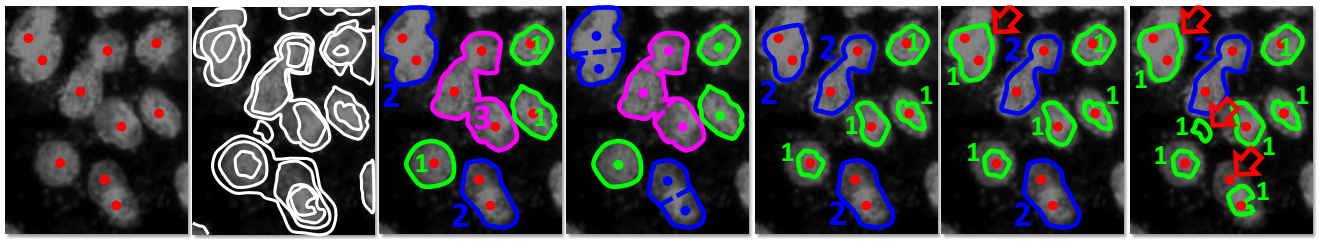
The objective of this work is to detect all the instances of an object of interest in microscopy images. The instances may be partially overlapping and clustered. To this end we introduce a tree-structured discrete graphical model that is used to select and label a set of non-overlapping regions in the image by a global optimization of a classification score. Each region is labelled with the number of instances it contains – for example regions can be selected that contain two or three object instances, by defining separate classes for tuples of objects in the detection process.

We show that this formulation can be learned within the structured output SVM framework, and that the inference in such a model can be accomplished using dynamic programming on a tree structured region graph. Furthermore, the learning only requires weak annotations – a dot on each instance. The candidate regions for the selection are obtained as extremal region of a surface computed from the microscopy image, and we show that the performance of the model can be improved by considering a proxy problem for learning the surface that allows better selection of the extremal regions. Furthermore, we consider a number of variations for the loss function used in the structured output learning.

The model is applied and evaluated over six quite disparate data sets of images covering: fluorescence microscopy, weak-fluorescence molecular images, phase contrast microscopy and histopathology images, and is shown to exceed the state of the art in performance.

Keywords:
Cell detection, microscopy image analysis, overlapping object detection

*Corresponding author
Email address: carlos.arteta@eng.ox.ac.uk (Carlos Arteta)



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

Figure 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.

1. Introduction

Automatic detection of objects (e.g. cell colonies, individual cells or nuclei) in microscopy images plays a crucial role in the analysis of microscopy-based experiments within a wide variety of microscopy applications for clinical and commercial settings. On its own, detection is able to determine the presence (and quantity) of an object of interest, such as cancer cells in a pathology image, but furthermore, it can also be the starting point for other objectives such as object segmentation or tracking. Among the challenges that characterise object detection in microscopy images, one that stands out is the necessity to deal with the presence of a large number of objects, often partially overlapping.

In many microscopy imaging modalities, objects of interest can often be identified as bright or dark blobs in one of the image channels. Such blobs correspond to *extremal regions* [22], and a natural approach to detection and understanding such images is (a) to consider the set of all extremal regions and (b) to identify those extremal regions that actually correspond to objects of interest. This is the approach that we pursue in this work. Several key challenges need to be addressed to make this approach successful, namely:

- Each object of interest typically corresponds to multiple, very similar and overlapping, extremal regions. The challenge then is to pick a subset of regions corresponding to objects of interest, so that each object of interest is represented by only one region. We show how this can be achieved via organizing extremal regions into a tree-shaped (or forest-shaped) discrete graphical model with binary labels. Message propagation (dynamic programming) in such a tree (forest) then produces a desired subset of *non-overlapping* extremal regions corresponding to objects.

- In more challenging images, it is often the case that groups of tightly overlapping objects (i.e. cells in a dense cluster) cannot be distinguished on the basis of extremal regions. In other words, for certain objects, there might not exist extremal regions that include one object

but exclude others in the same group. We show that the model can be extended to handle such challenging situations. The extended model is able to identify the blobs (extremal regions) that correspond to *multiple* overlapping objects, and to label simultaneously the selected regions with labels that indicate the number of objects that each selected extremal region corresponds to (Figure 1). This extension greatly widens the applicability of the approach without changing the topology of the underlying graphical model or increasing the complexity of the inference.

- Apart from the model and the inference in the model, a key question is one of machine learning, i.e. a method to identify which extremal regions correspond to objects and which do not (and in the extended case, identify the number of objects within certain regions). We demonstrate that all this can be done in a *weakly-supervised* learning setting, so that the method is trained on a set of *dotted* representative images, where each object is annotated only by a dot placed inside of it. The training is performed using latent structured output support vector machines [34] with a specially designed *counting* loss-function.

- Finally, we address the task of automatically identifying a "good" image channel that contains extremal regions that are "good" for our approach. Specifically, we propose a method that automatically optimizes over a linear combination of input channels (where some input channels can actually correspond to filtered versions of other channels) to determine an input image. After such an optimization, the resulting image gives rise to extremal regions that allows the efficient identification of individual objects or small groups of overlapping objects. This procedure only requires the same dot annotated images as above.

We conduct a set of experiments with synthetic and real microscopy images and show that the proposed method achieves very good detection accuracies despite large amounts of overlap, and very low effective spatial resolution. We assess the effect of the different elements of our detection system and show that the combination of them results in the highest accuracy. The resulting system outperforms other methods for instance detection in microscopy images, and is comparable in counting accuracy with the methods that are trained to count (and do not perform detection). While microscopic image modalities form a natural domain for our method, the proposed approach is general and can be applied to macroscopic medical or non-medical images, as demonstrated in [3].

This paper extends the previous conference papers [2, 3] that developed the initial approach. In comparison to the more recent conference version [3], this paper adds the following extensions: *(i)* it develops in more detail the inference procedure on the tree-structured graphical model (Section 5); *(ii)* it provides further evaluation and insights into the loss function, along with a new variant of it (Section 6.1); *(iii)* it proposes a method for picking a linear combination of input channels that optimizes the method's performance (Section 7); and *(iv)* it provides additional experiments with challenging microscopy images (Section 8).

2. Background

Instance detection in crowded scenes. Most computer vision methods that address the task of understanding images with multiple overlapping objects fall into two classes. The first is based on

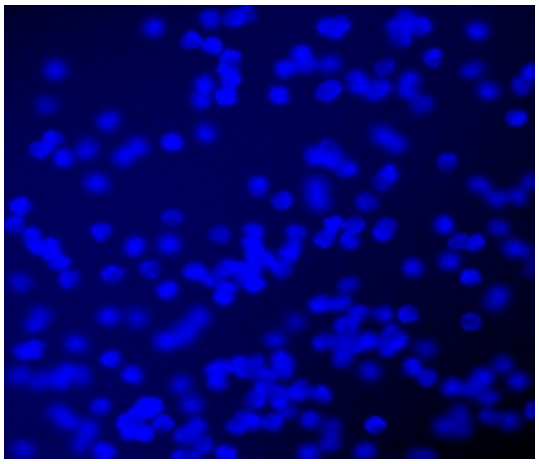
individual object detection. Such detection can be based on a sliding window or Hough transform, followed by an appropriate non-maxima suppression procedure [18, 9, 5], stochastic fitting of interacting particles or object models [11, 10, 33], or region-based detection [23, 25, 2]. The second class contains the methods that avoid the detection of individual instances but instead perform analysis based on local or global texture and appearance descriptors, e.g. by recovering the overall real-valued count of objects in the scene [15, 21, 7, 28] or by estimating the local real-valued density of the objects in each location of interest [20, 13].

Depending on the degree of overlap between objects, the first or the second class of methods might be more appropriate. 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.

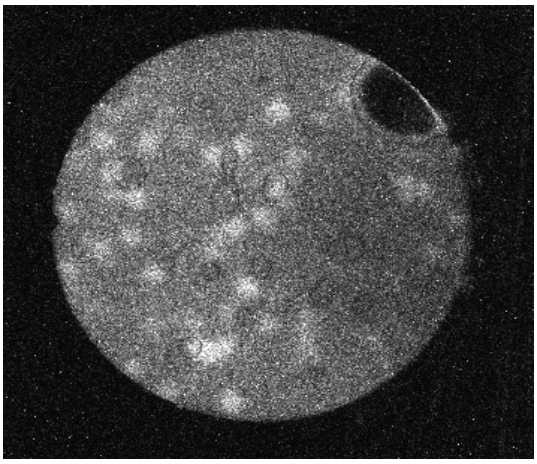
However, for the high-density images detection-based analysis may fail badly, especially when the amount of overlap and inter-occlusion between objects makes the detection of individual instances hard or impossible even for human experts. In such situations, the performance of density/global count estimation methods degrade more gracefully than detection-based 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.

In real life, many 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 example, a microscopy image may contain both regions of low and high cell density corresponding to different tissues or colonies at different stages of growth. Our proposed method aims to perform within such scenario, making the link between individual instance detection and instance counting in dense scenes.

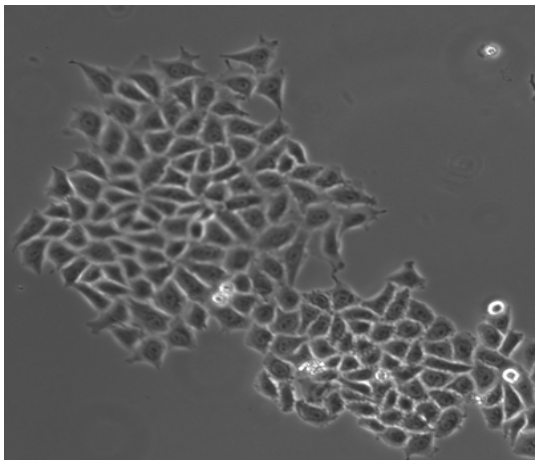
Biological object detection in microscopy images. It is often convenient to model biological objects (i.e. cells) in microscopy images as regions of local minima or maxima in the intensity channels, thus common blob detectors (or related custom algorithms) have commonly been applied for this instance detection task [1, 27, 30, 6, 16, 32]. A blob detector approach within the microscopy image analysis scenario also benefits from the fact that the common task can be seen as detection of multiple instance of the same object (i.e. all cells in the image of a cell culture will look similar). However, in non-trivial scenarios, such as cluttered images with cell overlap, or cases where cell discrimination is required, blob detectors do not have the flexibility to capture complex cell models, and thus, fail to achieve a good balance between sensitivity and specificity. Nevertheless, if they are sensitive enough, blob detectors can produce useful sets of candidates that can be further evaluated with more complex statistical models and this is demonstrated in this work.



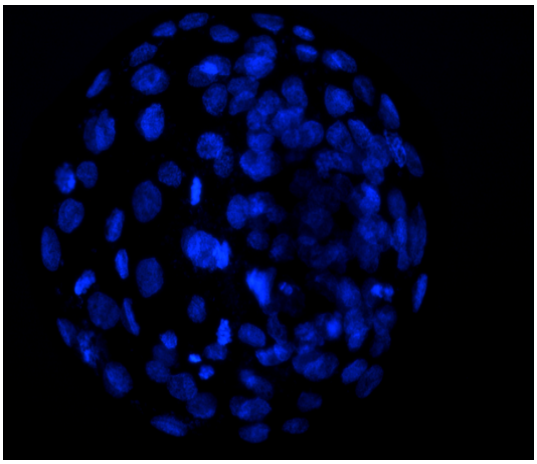
(a) Synthetic cells in fluorescence



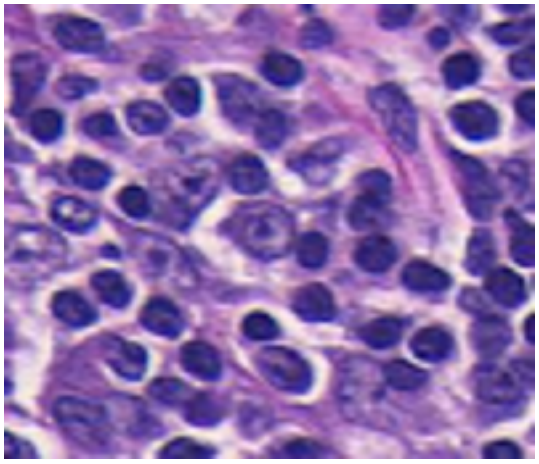
(b) Weak-fluorescence molecular imaging



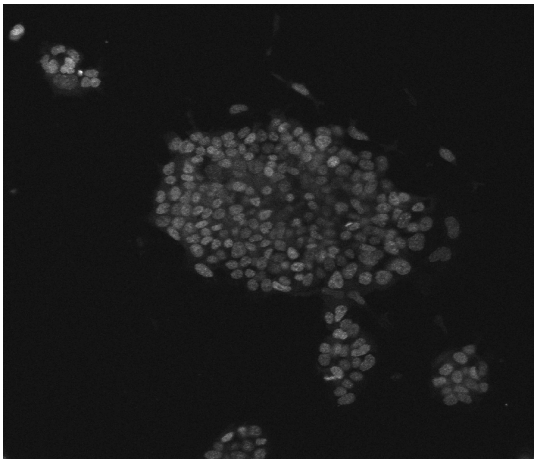
(c) HeLa in Phase contrast



(d) Blastocyst



(e) Histopathology



(f) Cell nuclei in fluorescence

Figure 2: Example images from the various microscopy datasets used throughout the paper. See Section 3 for details.

3. Datasets

We first introduce the datasets and metrics used to evaluate the performance of our system before describing the system itself. Six distinct microscopy datasets spanning different modalities and imaging conditions are used. 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 (i.e. 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 paper 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 2a). The synthetic dataset [20] represents a good benchmark for comparison of cell detection and counting methods as it contains perfect ground truth annotations due to its synthetic nature. It consists of 200 images of cell nuclei on fluorescence microscopy generated with [17]. 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 [20]. 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 2b). The molecular dataset consists of images of gels with DNA colonies obtained through *in vitro* amplification [8] (the method is also known as a "polony" technique [24]). 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 [29]. 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 2c). This dataset introduced in [2] 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. 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 [2].

Blastocysts (Figure 2d). 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 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 2e). The histopathology dataset was introduced in [14] 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 2f). The final dataset 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 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.

4. Model overview

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 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). More generally, we can transform the input image in various ways, creating a new map $\mathcal{J}$ where higher-value regions correspond to higher probabilities of an object's presence. The pool of candidate regions can then be generated as a set of extremal regions in the transformed image $\mathcal{J}$ (see Section 7) .

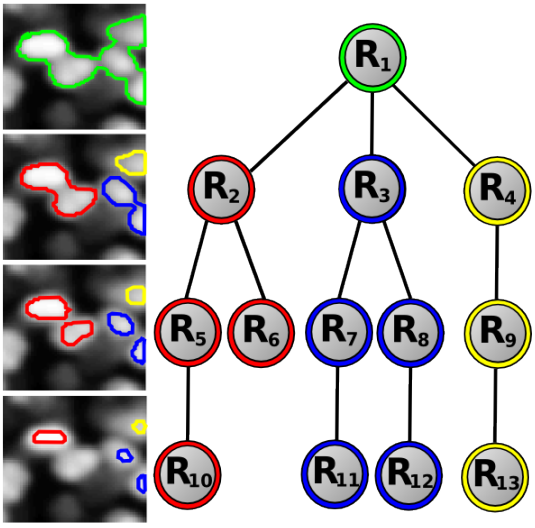


Figure 3: Typically, many extremal regions are nested within and between cells (especially when there is cell clumping) forming a tree structure. For example, the boundaries of several extremal regions that appear in the close-up of a cell image are shown, which can be represented by the tree structure. The parent-child relationships in the tree correspond to the nestedness of the regions. The tree structure is utilized by the inference algorithm. The colour-coding matches regions in the cell images with their corresponding node in the tree-structured graph.

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 6), 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 5) 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.

5. Inference on the model

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:

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

This constrained maximization thus simply tries to maximize the cumulative score of all selected regions.

The maximization of (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). 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.

We achieve this using the same trick as in [19]. We introduce the auxiliary variables z that are uniquely determined by the initial variables $\mathbf{y}$ in the following sense: $z_i = d > 0$ iff either $y_i = d$ or some y_k such that R_k is an ancestor of R_i in the tree equals d (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 (1) can then be rewritten as a pairwise tree-structured MRF on the auxiliary variables:

$$F(\mathbf{z}) = \max_{\mathbf{z}} \sum_{i|p(i)\neq 0} W_i(z_i, z_{p(i)}) + \sum_{i|p(i)=0} V_i(z_i), \tag{2}$$

where $p(i)$ maps region R_i to the number of its parent region ($p(i) = 0$ for root regions in the forest), $W_i(d, d) = 0$, $W_i(d, 0) = V_i(d)$, $W_i(0, d{>}0) = -\infty$, and $W_i(d_1, d_2 \neq d_1) = -\infty$ as long as $d_2 > 0$.

After such variable change, all $y \in Y$ are one-to-one mapped to z configurations with the finite values of the functional (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 some constant label d. This corresponds to the labeling that assigns $y_i{=}d$ 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 (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 (2) can be accomplished via tree-based dynamic programming [26] (the max-sum version of the algorithm). It is then trivial to compute the optimal solution of (1) from the optimal solution of (2).

6. 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 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)* (3), denoted as L_{IC}, that penalizes all deviations from the one-to-one correspondences between annotation dots and the selected regions (Figure 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, and D^j and N^j be the total number of dots and candidate regions in $\mathcal{I}^j$ respectively. The *IC-loss* imposed by such annotation on each possible region labeling $\mathbf{y}$ is formulated as:

$$L(\mathbf{y}^j) = \sum_{i=1}^{N^j} [y_i^j > 0] \Delta(d_i^j, y_i^j) + D^j - \sum_{i=1}^{N^j} [y_i^j > 0] d_i^j. \tag{3}$$

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 6.1. 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 (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 [31]. Thus, since the loss (3) 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:

$$\min_w ||\mathbf{w}||^2 + \frac{C}{M} \sum_{j=1}^{M} \max_{\mathbf{y}^j \in \mathcal{Y}^j} \left(L_{IC}(\mathbf{y}^j) + \mathbf{w} \cdot \left(\Psi(f^j, \mathbf{y}^j) - \Psi(f^j, \bar{\mathbf{y}}^j) \right) \right), \tag{4}$$

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 (6)) with zero IC-loss, and $\Psi(\mathbf{f}^j, \mathbf{y}^j)$ is the *joint feature representation* defined as follows:

$$\Psi(\mathbf{f}^j, \mathbf{y}^j) = \left[\begin{array}{c} \sum_{i=1}^{N^j} [\mathbf{y}^j = 1] \mathbf{f}_i^j, \ldots, \sum_{i=1}^{N^j} [\mathbf{y}^j = D] \mathbf{f}_i^j \end{array} \right]^T. \tag{5}$$

The optimization objective (4) can be minimized with a cutting plane algorithm [31], for which an efficient way of computing the most violated constraint is required. Specifically, we need to compute the second term of equation (4) for a fixed $\mathbf{w}$ (*loss-augmented inference*). Fortunately, in our case the loss (3) decomposes in an appropriate way, and the loss-augmented inference corresponds to the following optimization (after removing the terms independent from $\mathbf{y}^j$):

$$\max_{\mathbf{y}^j \in \mathcal{Y}^j} \sum_{i=1}^{N^j} [y_i^j > 0] \left(\Delta(d_i^j, y_i^j) - d_i^j \right) + \mathbf{w} \cdot \left(\Psi(f^j, \mathbf{y}^j) \right), \tag{6}$$

The maximization of (6) is then reduced to the optimization (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 1c,e). To handle this, we follow a conventional way [34] and add the "ground truth" configuration for each image into the optimization (4) 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:

$$\min_{\mathbf{w},\mathbf{h}^j\in\mathcal{H}^j} \left\{ ||\mathbf{w}||^2 + \frac{C}{M}\sum_{j=1}^{M} \max_{\mathbf{y}^j\in\mathcal{Y}^j} \left(L(\mathbf{y}^j) + \mathbf{w}\cdot\Psi(f^j,\mathbf{y}^j)\right) - \frac{C}{M}\sum_{j=1}^{M} \mathbf{w}\cdot\Psi(f^j,\mathbf{h}^j) \right\}. \tag{7}$$

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) 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 (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 (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 $N^j\, v$.

6.1. Penalization function for the IC-loss

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. 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.

We evaluate the variants of the penalization function $\Delta(\cdot,\cdot)$ shown in Table 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

$\Delta^u(d_i^j, y_i^j)$	$\lvert d_i^j - y_i^j \rvert$
$\Delta^{x^2}(d_i^j, y_i^j)$	$(d_i^j - y_i^j)^2$
$\Delta^s(d_i^j, y_i^j)$	$\lvert y_i^j - d_i^j \rvert / (d_i^j + 1)$
$\Delta^a(d_i^j, y_i^j)$	$\begin{cases} (y_i^j - d_i^j)/(d_i^j + 1), \text{ if } y_i^j \geq d_i^j \\ d_i^j - y_i^j, \text{ if } y_i^j \leq d_i^j \end{cases}$
$\Delta^g(D_i^j, y_i^j)$	$\begin{cases} (y_i^j - D_i^j)/(D_i^j + 1), \text{ if } y_i^j \geq D_i^j \\ D_i^j - y_i^j, \text{ if } y_i^j \leq D_i^j \end{cases}$, 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 1: **Variants of the penalization function $\Delta(\cdot, \cdot)$.**

inside a candidate region (d_i^j), we adopt the principle of "smoothed" dot-annotations of [20]. 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 2a), for the splits of $N = 32$: five draws of 32 training images and 32 validations images (see Section 3 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 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 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.

6.2. Implementation details

Postprocessing 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 postprocessing in this case. For each selected region R_i we run k-means with $k = y_i$ on the

	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 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 1 and the text for the definitions of the functions.

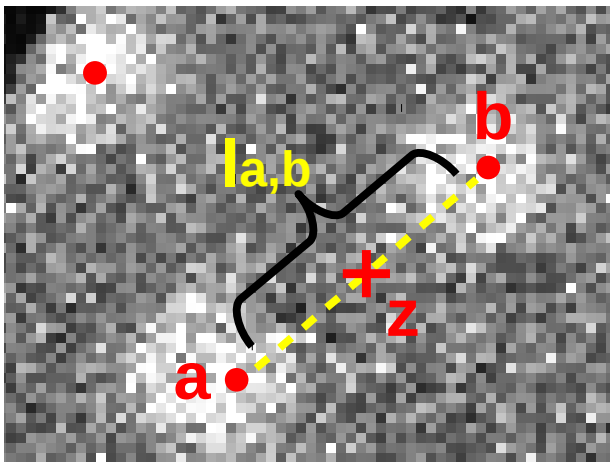


Figure 4: The intuition behind the surface optimization (8) is that we want to collect extremal regions on a surface that is higher (by a margin) in the dot-annotations $\mathbf{a}$ and $\mathbf{b}$ than it is in a point $\mathbf{z}$ between them. $\mathbf{z}$ is a latent variable which indicates a location within the line $l_{a,b}$ connecting a pair of dot-annotations. See text for more detail.

image coordinates of all pixels in that region, thus obtaining an estimate for the set of centroids of individual objects. An example of this postprocessing is shown in Figure 1(d).

Initialization and termination for learning. The initialization of $\mathbf{w}$ for the alternation-based maximization (7) 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.

7. Crafting a surface for extremal region computation

Collecting extremal regions as candidates for object detection from the intensity channel of microscopy images is often successful [2], but not optimal. For example, images with high levels of noise (i.e. weak-fluorescence images – Figure 2b), low contrast or images with highly inhomogeneous objects 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 corre-

spond to object groups. Nevertheless, for such cases, we show 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. Throughout the paper, we refer to the height map defined over the generated 2D image channel as a *surface*. The computation of this surface can be done as a preprocessing step that is independent from other parts of the system.

Here, we propose to compute a surface optimised 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 Section 4 and are assumed to mostly lie within the objects of interest. Let $\mathbf{a}$ and $\mathbf{b}$ be the dot-annotations for two neighbouring instances of an object in our images, and $\mathbf{z}$ be a point between them whose selection is described below (see Figure 4). We want the surface S^j to be higher in $\mathbf{a}$ and $\mathbf{b}$ than it is in $\mathbf{z}$ 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 $F(a)$, $F(b)$ and $F(z)$ 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

$$F^j(a) = \begin{pmatrix} X_1^j(a_1) & X_1^j(a_2) & \dots & X_1^j(a_D) \\ X_2^j(a_1) & X_2^j(a_2) & & X_2^j(a_D) \\ \vdots & & \ddots & \vdots \\ X_N^j(a_1) & X_N^j(a_2) & & X_N^j(a_D) \end{pmatrix}$$

Therefore, $S^j(a) = \alpha^T F^j(a)$ 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 $F(a) = [F^1(a), F^2(a), \dots, F^J(a)]$. $F(a)$, $F(b)$ and $F(z)$ are used to easily compute the margin violations within the constraints of the optimization (8), where one slack variable $\xi_{a,b}$ is introduced for every pair $\mathbf{a}$ and $\mathbf{b}$ of dot-annotations.

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 L^j containing the standard deviation of the Laplacian of each feature channel: $L^j = [\sigma(\triangle X_1^j), \sigma(\triangle X_2^j), \dots, \sigma(\triangle X_N^j)]^T$. For the entire training set $\mathcal{I}$, we compute a single vector L as the mean standard deviation of the corresponding feature channels.

Finally, we find α through the minimization

$$\begin{aligned} \min_{\alpha} \quad & \alpha^T L + \lambda \sum_{\forall a,b \in \mathcal{D}} \xi_{a,b} \\ \text{s.t.} \quad & \alpha^T (F(a) - F(z)) \geq 1 + \xi, \\ & \alpha^T (F(b) - F(z)) \geq 1 + \xi, \\ & \xi \succeq 0, \alpha \succeq 0. \end{aligned} \tag{8}$$

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 (8). 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 $\mathbf{z}$. The variable $\mathbf{z}$ corresponds to the location between every pair of dot-annotations which would serve as reference for the optimization in (8). 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 (8) 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 4). 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 (8), 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.

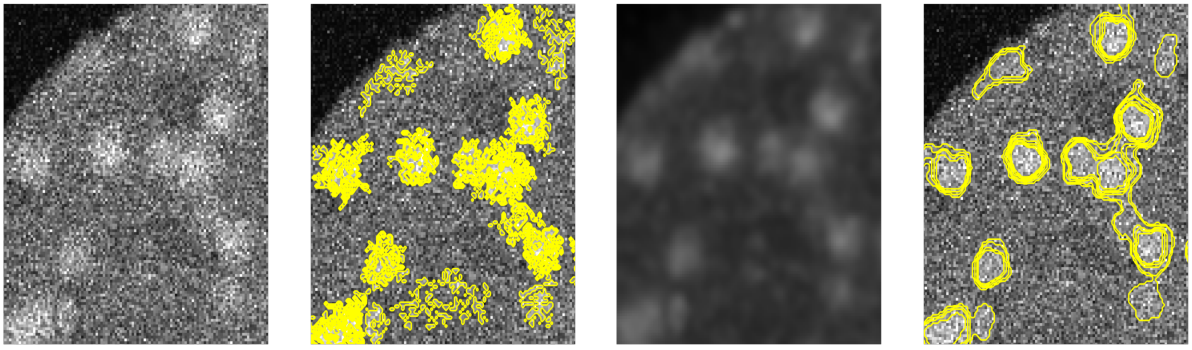
7.1. Validation experiments

In order to demonstrate the usefulness of the surface optimization, we assess the performance of the model on the weak-fluorescence molecular dataset (Figure 2b) with and without this pre-processing step.

Qualitatively, it can be seen (Figure 5) that the surface optimization procedure has two positive effects: first, due to the smoothness enforced on the surface (Figure 5c), the pool of candidate regions (Figure 5d) 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 5b); 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 3, the surface computation

	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 3: **Evaluation of the effect produced by the computation of candidate regions on an optimized surface.** The evaluation is done on the molecular dataset (Figure 2b).



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

Figure 5: 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.

on the molecular dataset leads to higher detection accuracy and lower computation time per image due to the reduced number of candidate regions.

8. Experiments and results

We now evaluate the performance of the model on the datasets described in Section 3. Within these experiments, the *full system* refers to the method described in this paper using the penalization function Δ^g (see Section 6.1) and the surface learning (see Section 7). The usage of the optimized surface, however, is determined via cross-validation. Thus, the surface is not enabled for datasets where it is not beneficial. Instead, extremal regions are collected directly from the intensity channel of the image. The reasons for discarding the surface learning are discussed within the analysis of the experiments.

The full system is compared in all cases against its single class version (i.e. as presented in [2]), denoted as *singletons*. Additionally, we show the performance of the single class version combined with the surface optimization (*singletons w/ surface*) when the latter is required. Finally, we show results of other detection and counting methods when available (i.e. for the publicly available datasets).

The hyperparameters of the model (e.g. C in (7)) are learnt via cross-validation using the appropriate measure. That is, the precision and recall values shown in this section correspond

to the combination of hyperparameters that produced the highest F_1-score on the validation sets, whereas MCE values correspond to those hyperparameters that produced the lowest mean counting error.

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 characterise the size, shape, colour (or intensities) and information regarding the local context of regions. The specific dimensions of the descriptors where chosen empirically, but the method is not sensitive to such choices. In detail, a feature vector concatenates the following descriptors: (i) a 150-dimensional vector soft-encoding region size by setting to 1 the entry corresponding to the size of the region, and then smoothing the resulting binary vector with a Gaussian kernel, (ii) a 12-dimensional histogram of intensities inside the region, (iii) two 8-dimensional histogram of difference of intensities between the boundary of the extremal region and a dilation of it (over two different dilation scales), (iv) a shape descriptor represented by a 60-dimensional histogram of the distribution of the boundary of the region on a size-normalized polar coordinate system, and finally, (v) a binary vector of the same dimension as the number of classes which encodes the number of leaf regions (i.e. regions without nested regions in the pool) nested within a given region. This last descriptor often indicates the presence of individual objects existing inside the region being encoded.

Finally, the parameters for the computation of the $MSERs$ are maintained for all of the experiments.

8.1. Synthetic cells

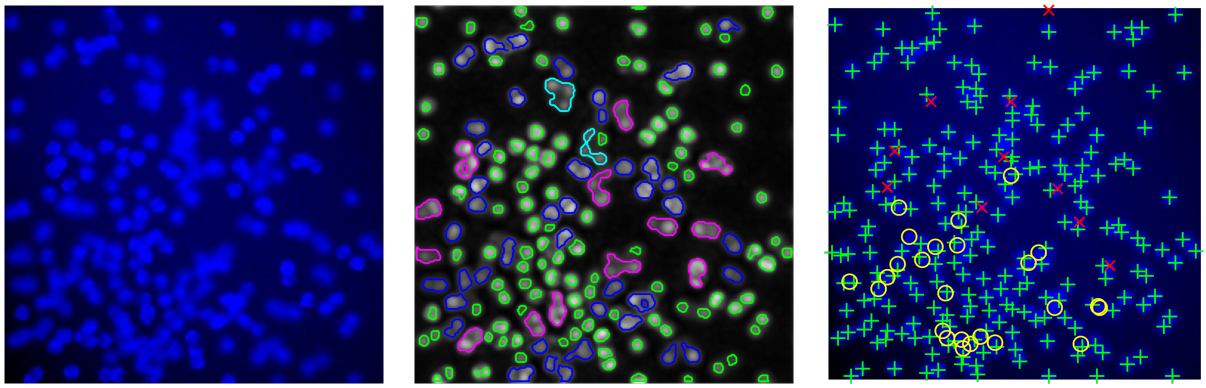
We perform the evaluation over this dataset using the splits of $N = 32$ proposed in [20], which consist of 5 different splits with 32 images for training and 32 for validation. Results are presented in Table 4, and an example can be seen in Figure 6.

The high cell confluency in the synthetic cell dataset poses a difficult challenge for detection algorithms due to very high cell overlap. Therefore, it is expected that counting algorithms such as [13, 20] would outperform detection methods. Nonetheless, our method is able to produce a comparable mean counting error (MCE), while providing estimates of object localization evaluated with precision and recall. The single class version of our method is unable to detect objects in dense groups, and thus fails badly in this dataset. The extension to multiple classes (tuples) allows the method to handle cell clusters and boosts the recall of the detection, especially when the penalization in the cost function is re-scaled to compensate for the high intraclass variability of high order classes (see Section 6.1).

We note that the surface optimization was ruled out of the evaluation on this dataset during cross-validation due to the reason explained next. The surface learning breaks clusters into smaller parts, as expected. However, when this occurs within dense clusters of high-order class (i.e. with more than 5 heavily overlapping cells) the number of resulting elements tends to be less than the number of instances that the cluster originally contained. When such smaller elements are parsed as lower order classes, the benefit of the rescaled cost function (Δ^g) for handling high-order classes is then diminished, resulting in a considerable reduction of the overall detection recall. Nevertheless, we argue that the existence of such high-order clusters with heavily overlapping (and indistinguishable) instances is an artifact created by the synthetic nature of this dataset and we did not encounter such extreme cases in real microscopy images.

	MCE	Prec.	Rec.	F_1-score
Fiaschi *et al.* [13]	3.2 ± 0.1	-	-	-
Lempitsky & Zisserman [20]	3.5 ± 0.2	-	-	-
Barinova *et al.*[4]	6.0 ± 0.5	-	-	-
Singletons	51.2 ± 0.8	98.87 ± 1.52	72.07 ± 0.85	83.37 ± 1.20
Full system w/o surface	5.06 ± 0.2	95.00 ± 0.75	91.97 ± 0.43	93.46 ± 0.15

Table 4: **Detection and counting accuracy for the synthetic cell dataset (split N=32).** Please see text Section 8.1 for details.



Original Selected Regions Detection

Figure 6: Example detection result in the synthetic dataset. Selected regions (*middle*) are colour-coded according to the number of instances they contain: green = 1, blue = 2, magenta = 3 and cyan = 5. In the detection image (*right*), correct detections are denoted with a green '+', false detections with a red 'x' and missed instances with a yellow '$\circ$'.

8.2. Molecules in fluorescence microscopy

Within our experiments, the molecular dataset shows the greatest benefit of the surface 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 of the refinement of the candidate region pool. Results are presented in Table 5, and an example can be seen in Figure 7.

	MCE	Prec.	Rec.	F_1-score
Singletons	15.59 ± 0.48	88.14 ± 1.75	41.19 ± 1.78	56.11 ± 1.51
Singletons w/ surface	6.88 ± 0.50	84.01 ± 2.59	69.75 ± 1.54	76.20 ± 1.61
Full system	7.12 ± 0.36	80.01 ± 3.62	75.09 ± 2.17	77.43 ± 1.98

Table 5: **Detection and counting accuracy for the molecular dataset.** See Section 8.2 for details.

8.3. HeLa in phase contrast microscopy

With relatively limited cell overlap and mostly well-defined cell boundaries, experiments on the phase contrast microscopy dataset showed the least benefit from the additional components of our system over the single class version. For example, the full system mostly produces singleton

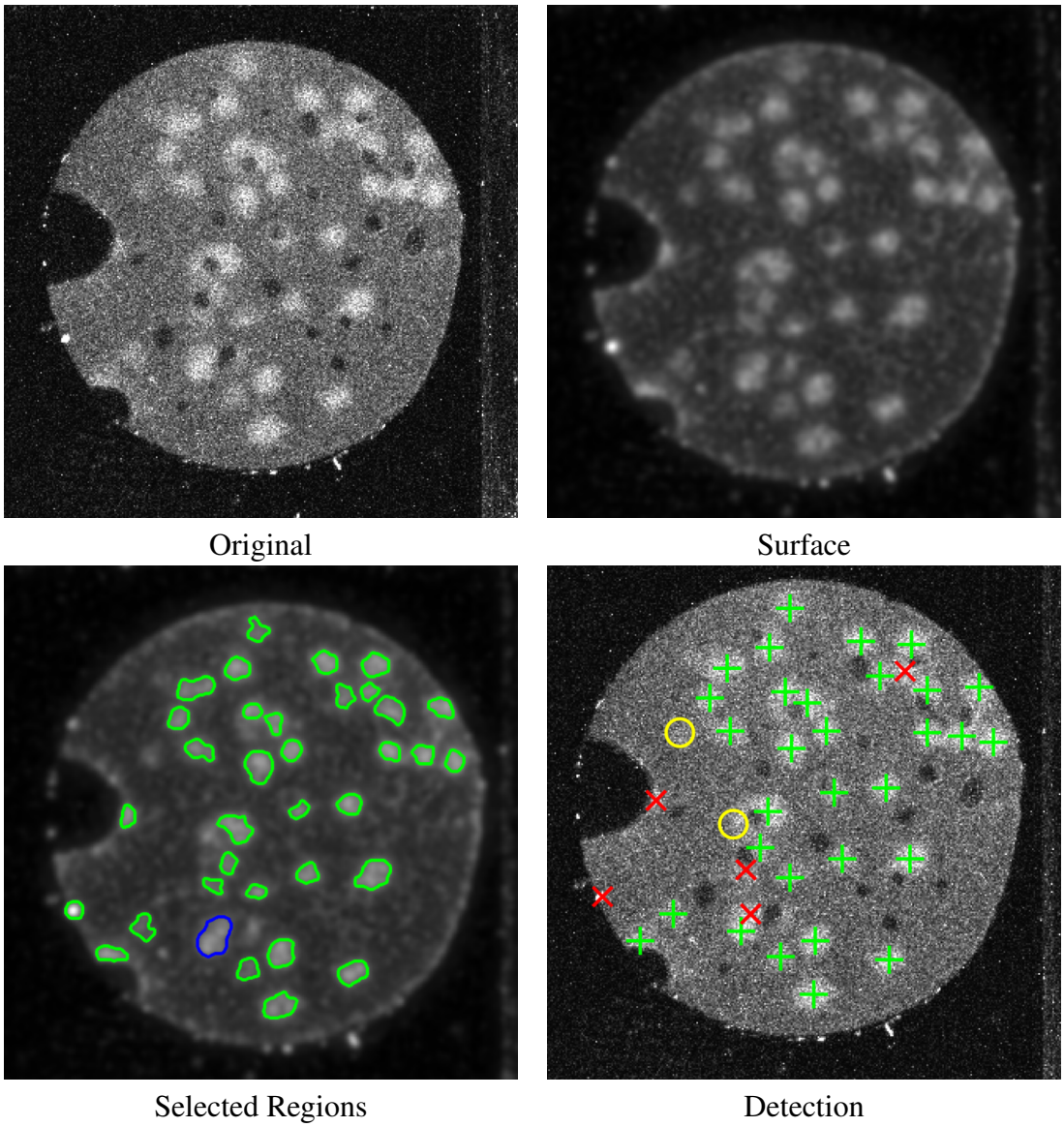
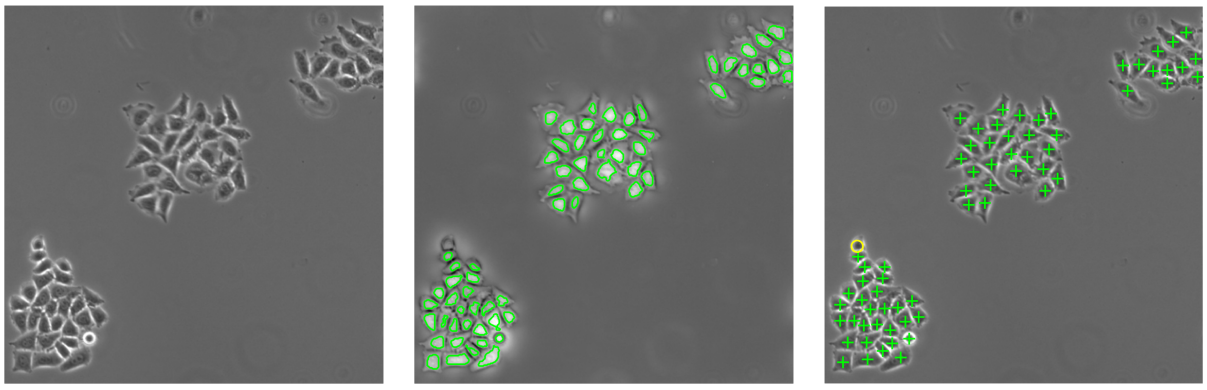


Figure 7: Example detection result in the dataset of molecular imaging with weak-fluorescence. Selected regions (*bottom left*) are colour-coded according to the number of instances they contain: green = 1 and blue = 2. In the detection image (*bottom right*), correct detections are denoted with a green '+', false detections with a red 'x' and missed instances with a yellow '○'.



Original Selected Regions Detection

Figure 8: Example detection result for HeLa cells in phase contrast microscopy. All the selected regions (*middle*) in this image correspond to singletons. In the detection image (*right*), correct detections are denoted with a green '+', false detections with a red 'x' and missed instances with a yellow '○'.

detection and the surface optimization was discarded during cross-validation. Nevertheless, the experiments show the high accuracy that can be achieved in this microscopy modality when using extremal regions as candidates (along with the appropriate region evaluation and selection model).

For this dataset, we have added the detection result of a recent method for cell detection and segmentation based on correlation clustering [35] (authors' implementation). The latter achieves a high detection accuracy with the added benefit of optimizing for cell segmentation. Nevertheless, it is outperformed in the detection task by the full version of our system. Even though the raw output of our method are regions which can often match the boundaries of the objects, we do not compare segmentation metrics as our method does not optimize segmentation masks.

The results are presented in Table 6, and an example can be seen in Figure 8.

	MCE	Prec.	Rec.	F_1-score
Correlation clustering [35]	-	-	-	95
Singletons	2.36 ± 0.67	93.70 ± 0.20	91.94 ± 0.72	92.81 ± 0.35
Full system w/o surface	3.84 ± 1.44	98.51 ± 1.16	95.76 ± 0.27	97.10 ± 0.27

Table 6: **Detection and counting accuracy for the phase contrast dataset of [2].** See Section 8.3 for details.

8.4. Blastocysts

We found the performance of our system on the challenging blastocysts dataset to significantly benefit from both the surface optimization and the handling of overlapping regions. The latter was expected due to the large amount of cell overlap present in this dataset, resulting from the projection of the blastocyst sphere (3D) into a 2D image. Results are presented in Table 7, and an example can be seen in Figure 9.

8.5. Cell nuclei in fluorescence microscopy

The assessment of our detection method on the cell nuclei dataset showed that the detection accuracy (F_1-score) remained similar across the different variants of the method, but the mean

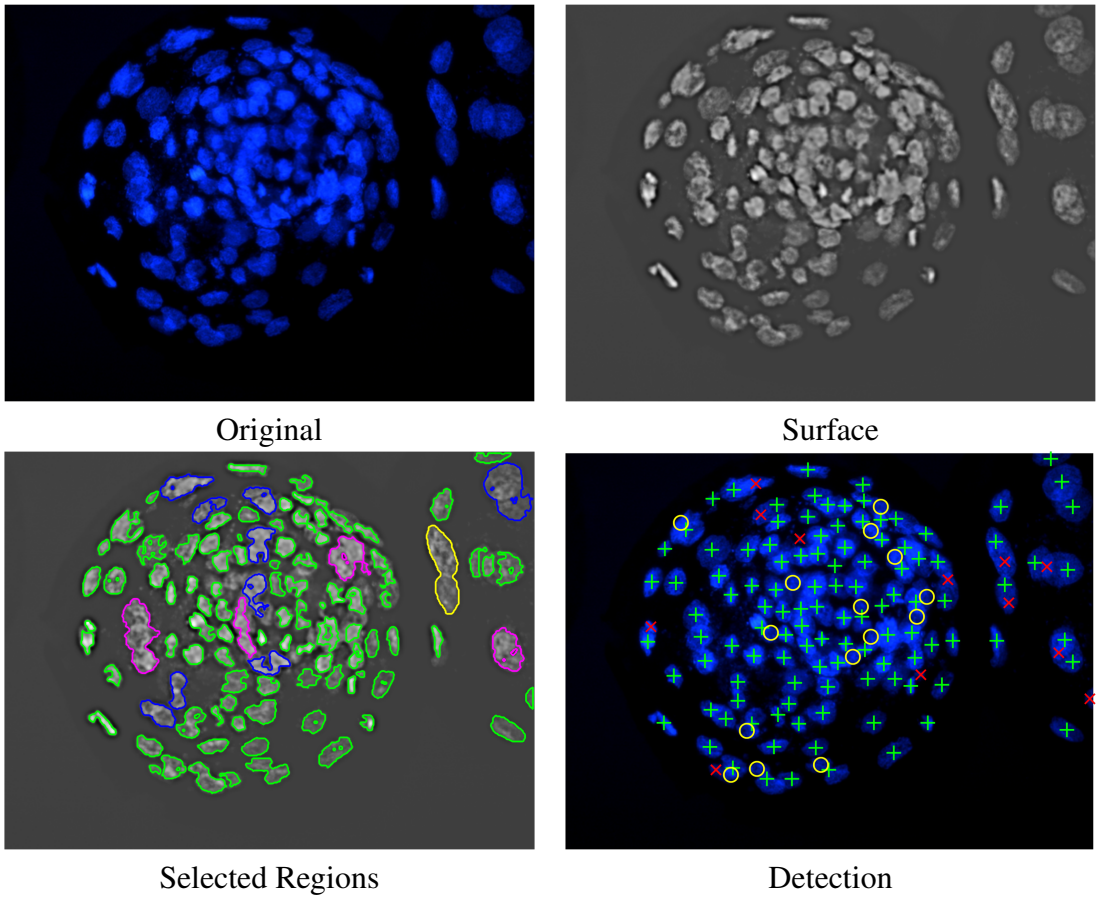


Figure 9: Example detection result in the blastocyst dataset. Selected regions (*bottom left*) are colour-coded according to the number of instances they contain: green = 1, blue = 2, magenta = 3 and yellow = 4. In the detection image (*bottom right*), Correct detections are denoted with a green '+', false detections with a red 'x' and missed instances with a yellow '○'.

	MCE	Prec.	Rec.	F_1-score
Singletons	37.79 ± 1.11	97.51 ± 0.83	62.84 ± 2.49	76.39 ± 1.63
Singletons w/ surface	25.35 ± 2.03	94.59 ± 0.29	72.87 ± 1.29	82.31 ± 0.78
Full system	9.24 ± 1.52	90.47 ± 1.00	81.77 ± 1.18	85.90 ± 0.94

Table 7: **Detection and counting accuracy for the blastocyst dataset.** See Section 8.4 for details.

counting error benefited from the extremal region collection over the optimized surface as opposed to the intensity channel. The reason is that the surface optimization managed to enhance the contrast on image areas that would be otherwise missed by the candidate region detection. Thus, the detection recall increased. Results are presented in Table 8, and an example can be seen in Figure 10.

	MCE	Prec.	Rec.	F_1-score
Singletons	46.82 ± 2.49	93.71 ± 0.23	81.74 ± 0.50	87.32 ± 0.19
Singletons w/ surface	16.90 ± 1.83	89.57 ± 1.10	88.48 ± 0.83	89.01 ± 0.19
Full system	20.42 ± 4.10	87.12 ± 1.17	91.10 ± 0.75	89.05 ± 0.29

Table 8: **Detection and counting accuracy for the dataset of cell nuclei in fluorescence microscopy.** See Section 8.5 for details.

8.6. Lymphocytes in histopathology images

As in the phase contrast dataset, the dataset for detection lymphocytes on breast cancer tissue [14] does not present significant cell overlap and individual instances can be selected through blob detection. However, the presence of breast cancer cells with very similar appearance (i.e. same staining) and under a low effective spatial resolution, increases the difficulty of the detection task. For example, the method of [16] suffers in this case from relying on the properties of the staining to discern between similar elliptical blobs. On the other hand, our method uses additional discriminative features which increase the detection accuracy. Results are presented in Table 9, and an example can be seen in Figure 11.

	MCE	Prec.	Rec.	F_1-score
LIPSyM [16]	-	70.08	70.21	69.84
Singletons	3.7 ± 2.05	85.89 ± 1.21	89.90 ± 0.98	87.85 ± 1.13
Full system w/o surface	3.9 ± 2.65	84.09 ± 1.65	91.06 ± 1.5	87.40 ± 1.66

Table 9: **Results for the dataset of the ICPR 2010 Pattern Recognition in Histopathological Images contest [14].** See Section 8.6 for details.

8.7. Results analysis

The experiments in this section show a wide variety of scenarios for object detection in microscopy images, where the benefits and limitations of the different elements of our proposed model have been shown.

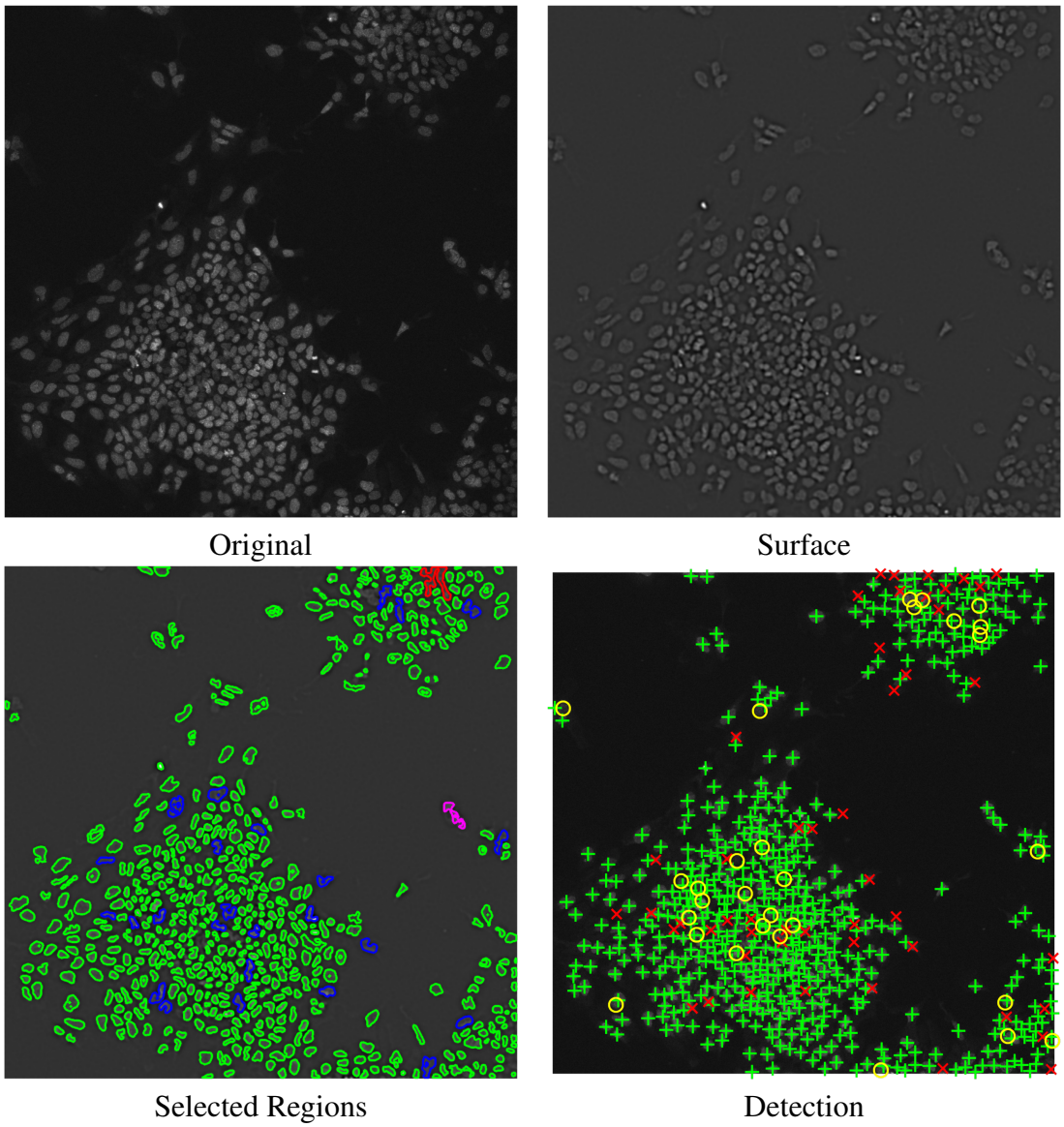


Figure 10: Example of cell nuclei detection in fluorescence microscopy. Selected regions (*bottom left*) are colour-coded according to the number of instances they contain: green = 1, blue = 2, magenta = 3 and red = 7. In the detection image (*bottom right*), correct detections are denoted with a green '+', false detections with a red 'x' and missed instances with a yellow '○'.

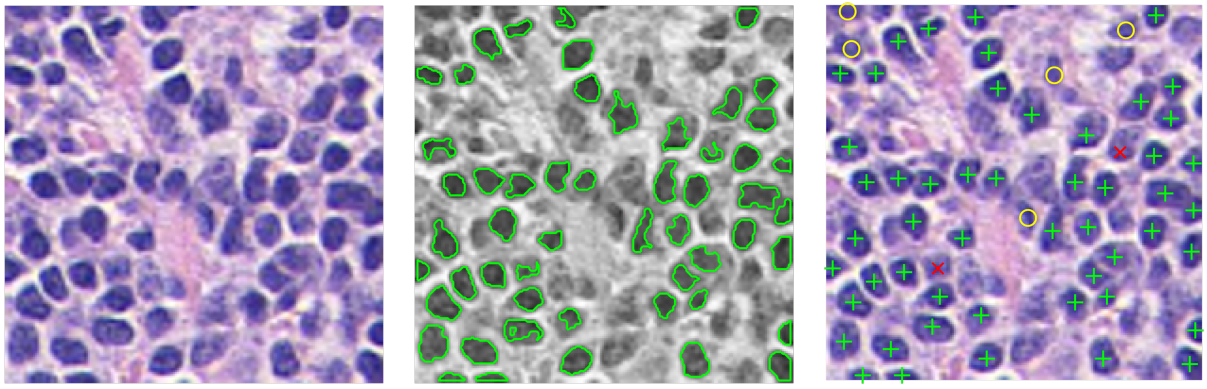


Figure 11: Example detection result for the dataset of the ICPR 2010 Pattern Recognition in Histopathological Images contest [14]. Selected regions (*middle*) in this image correspond to singletons. In the detection image (*right*), correct detections are denoted with a green '+', false detections with a red 'x' and missed instances with a yellow '$\circ$'.

The first aspect to emphasize is the applicability of extremal region trees as candidates for object detection in microscopy images, as initially observed in [2]. We have found that amongst the pool of candidates in the different trees, we can often find regions that provide a good delineation of the objects of interest from where strong features can be computed. This can be confirmed by inspecting the selected regions on the result images and noting that in most cases the method produces a fairly good segmentation of the objects regardless of the fact that only dot annotations are being used, and no post-processing of the regions is being done at any stage.

In terms of detection accuracy, for datasets with limited amount of object overlap, even the model limited to singleton detection would show a decent performance (i.e. F_1-score higher than 87% in histopathology, fluorescence and HeLa on phase contrast datasets). As overlap is introduced, the singleton-based model is still able to keep a high precision but quickly drops its recall. Intuitively, this comes from the fact that clusters of objects will be discarded during the evaluation for not fitting the model of the single object. When enabled to detect tuples of objects, the model can usually find a much better balance between precision and recall on datasets with object overlap, which translates into higher F_1-score and lower counting errors (e.g. in synthetic cells and blastocyst).

The introduction of the surface learning for the computation of extremal regions was found to mainly benefit extremal region collection for two conditions: noise and poor contrast. In the case of noisy images, best represented in this paper by the molecular dataset, the surface learning mainly helps by appropriately smoothing the objects. On the smooth surface, we observe more uniform candidate regions which better delineate the objects, providing better candidates to learn from, and to select from at test time. The case of poor contrast is best represented in this paper by the dataset of cell nuclei in fluorescence microscopy. Although not an extreme case, these images contain regions where cell nuclei are slightly out of focus and become much harder to distinguish. Nevertheless, due to the contrast enhancement term in the surface optimization, out-of-focus nuclei become easier to differentiate by the extremal regions, aiding the recall even for the

singleton baseline model. On the negative side, the surface was found to be harmful in cases with extreme overlap. Specifically, in the synthetic cell dataset where most of the objects are severely (and unrealistically) overlapping, the surface learning would break high-order clusters into smaller cluster with the appearance of singletons, negatively affecting recall. Therefore, the usage of the surface learning would be discarded during validation. Finally, we found cases where extracting the candidate regions directly from the image was good enough due to limited overlap and fairly homogeneous objects, and thus, the surface learning made no contribution (e.g. phase contrast and histopathology datasets).

The time required to train the method for each of our datasets is in the range of hours on a standard desktop computer (Intel i7 processor), while testing on a new image usually requires seconds. The bottleneck at testing time is in the feature computation, as each candidate region (usually slightly over a thousand per image) needs to be encoded individually. Nevertheless, the encoding of the regions is done in parallel, and thus, testing time reduces as more processing cores are used.

Overall, the most time consuming step of the end-to-end pipeline is the annotation of a training set, and the number of annotated images required to achieve a good performance will vary depending on the difficulty of the case. Nevertheless, we note that in simple cases very few images on training set can be sufficient to learn a reliable classifier. For example, high accuracy on the phase contrast dataset was achieved with a training set of only 11 images.

Finally, we note that although the methodology to compute cell centers from detected regions through k-means is quite simplistic, and considering that the F_1-score is based on cell centroid matching, we did not encounter examples where the performance of the method was harmed by faulty cell centre estimations given the criteria for the valid detection radius.

9. Summary and conclusion

We have presented a method for object detection in microscopy images (extending [2, 3]) which is particularly suitable for images with multiple overlapping instances of an object. 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.

To handle particularly challenging scenarios such as detection on noisy microscopy imaging modalities, we included 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.

We note that the method is not restricted to the usage of extremal regions as candidates. In practice, 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, 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 [12]), 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.

The proposed 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.

Acknowledgements. We acknowledge the researchers of the Laboratory for Viral RNA Biochemistry, Institute of Protein Research RAS, for providing the images and annotations of gels with molecular colonies, Dr. Svetlana Uzbekova (INRA, Physiology of Reproduction and Behavior Unit, Nouzilly, France) for providing 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) for proving the equipment and samples for the collection of the phase contrast dataset, Dr. Nasir Rajpoot for providing the histopathology dataset, and Dr. Julian Gingold for providing the dataset of cell nuclei in fluorescence microscopy. Financial support was provided by the RCUK Centre for Doctoral Training in Healthcare Innovation (EP/G036861/1) and ERC grant VisRec no. 228180.

References

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

[2] Arteta, C., Lempitsky, V., Noble, J. A., Zisserman, A., 2012. Learning to detect cells using non-overlapping extremal regions. In: Ayache, N. (Ed.), International Conference on Medical Image Computing and Computer Assisted Intervention. Lecture Notes in Computer Science. MICCAI, Springer, pp. 348–356.

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

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

[5] Barinova, O., Lempitsky, V., Kohli, P., 2010. On the detection of multiple object instances using Hough transforms. In: Proc. CVPR.

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

[7] Chan, A., Vasconcelos, N., 2009. Bayesian poisson regression for crowd counting. In: Proc. CVPR.

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

[9] Desai, C., Ramanan, D., Fowlkes, C., 2009. Discriminative models for multi-class object layout. In: Proc. ICCV.

[10] Descombes, X., Minlos, R., Zhizhina, E., 2009. Object extraction using a stochastic birth-and-death dynamics in continuum. Journal of Mathematical Imaging and Vision 33 (3), 347–359.
[11] Dong, L., Parameswaran, V., Ramesh, V., Zoghlami, I., 2007. Fast crowd segmentation using shape indexing. In: Proc. ICCV.
[12] Donoser, M., Bischof, H., 2006. segmentation by maximally stable volumes (msvs).
[13] Fiaschi, L., Nair, R., Köethe, U., Hamprecht, F., 2012. Learning to count with regression forest and structured labels. In: Proc. ICPR.
[14] Gurcan, M. N., Madabhushi, A., Rajpoot, N., 2010. Pattern recognition in histopathological images: An icpr 2010 contest. In: Recognizing Patterns in Signals, Speech, Images and Videos. pp. 226–234.
[15] Kong, D., Gray, D., Tao, H., 2006. A viewpoint invariant approach for crowd counting. In: Proc. ICPR.
[16] Kuse, M., Wang, Y., Kalasannavar, V., Khan, M., Rajpoot, N., et al., 2011. Local isotropic phase symmetry measure for detection of beta cells and lymphocytes. Journal of Pathology Informatics 2 (2), 2.
[17] Lehmussola, A., Ruusuvuori, P., Selinummi, J., Huttunen, H., Yli-Harja, O., 2007. Computational framework for simulating fluorescence microscope images with cell populations. IEEE TMI 29 (7), 1010–1016.
[18] Leibe, B., Leonardis, A., Schiele, B., 2008. Robust object detection with interleaved categorization and segmentation. IJCV.
[19] Lempitsky, V., Vedaldi, A., Zisserman, A., 2011. A pylon model for semantic segmentation. In: NIPS.
[20] Lempitsky, V., Zisserman, A., 2010. Learning to count objects in images. In: NIPS.
[21] Marana, A., Velastin, S., Costa, L., Lotufo, R., 1997. Estimation of crowd density using image processing. In: Image Processing for Security Applications, IEE Colloquium on.
[22] Matas, J., Chum, O., Urban, M., Pajdla, T., 2004. Robust wide-baseline stereo from maximally stable extremal regions. Image and Vision Computing 22 (10), 761–767.
[23] Matas, J., Zimmermann, K., 2005. A new class of learnable detectors for categorisation. In: SCIA.
[24] Mitra, R. D., Church, G. M., 1999. In situ localized amplification and contact replication of many individual dna molecules. Nucleic Acids Research 27 (24), e34–e39.
[25] Neumann, L., Matas, J., 2011. Text localization in real-world images using efficiently pruned exhaustive search. In: ICDAR.
[26] Pearl, J., 1988. Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference. Morgan Kauffman, California.
[27] Peng, H., Zhou, X., Li, F., Xia, X., Wong, S. T. C., 2009. Integrating multi-scale blob/curvilinear detector techniques and multilevel sets for automated segmentation of stem cell images. In: ISBI.
[28] Ryan, D., Denman, S., Fookes, C., Sridharan, S., 2009. Crowd counting using multiple local features. In: Proc. DICTA.
[29] Samatov, T. R., Chetverina, H. V., Chetverin, A. B., 2006. Real-time monitoring of dna colonies growing in a polyacrylamide gel. Analytical biochemistry 356 (2), 300–302.
[30] Smith, K., Lepetit, V., 2008. General constraints for batch multiple-target tracking applied to largescale videomicroscopy. In: CVPR.
[31] Tsochantaridis, I., Hofmann, T., Joachims, T., Altun, Y., 2004. Support vector machine learning for interdependent and structured output spaces. In: Proc. ICML.
[32] Wienert, S., Heim, D., Saeger, K., Stenzinger, A., Beil, M., Hufnagl, P., Dietel, M., Denkert, C., Klauschen, F., 2012. Detection and segmentation of cell nuclei in virtual microscopy images: a minimum-model approach. Scientific reports 2, 503.
[33] Wu, B., Nevatia, R., 2009. Detection and segmentation of multiple, partially occluded objects by grouping, merging, assigning part detection responses. IJCV 82 (2), 185–204.
[34] Yu, C., Joachims, T., 2009. Learning structural SVMs with latent variables. In: Proc. ICML.
[35] Zhang, C., Yarkony, J., Hamprecht, F., 2014. Cell detection and segmentation using correlation clustering. In: Medical Image Computing and Computer-Assisted Intervention MICCAI 2014. Vol. 8673.