



TECHNISCHE
UNIVERSITÄT
DRESDEN

BIOIMAGE METHODS FOR TRACKING CELLS DURING DEVELOPMENT

DISSERTATION
zur Erlangung des akademischen Grades
Doctor of Philosophy
(Ph.D.)

vorgelegt dem
Bereich Informatik
der Technischen Universität Dresden

von
COLEMAN WALKER BROADDUS
* 8. September 1987, Prince William, Virginia, USA

GUTACHTER: Prof. Eugene Myers
Max-Planck-Institut für Molekulare Zellbiologie und Genetik

Dr. Jan Funke
HHMI Janelia Research Campus

EINGEREICHT AM: 17. Jan 2023

COLEMAN WALKER BROADDUS
Bioimage Methods for Tracking Cells During Development

ABSTRACT

DETECTION, segmentation and tracking of cells and nuclei in images of biological systems are some of the key rate limiting steps in modern life science research. Automated approaches to these tasks promise many benefits, including accelerated research as well as reproducible data analysis, but so far computer methods are both insufficiently accurate and overly specialized to serve as a substitute for detailed human curation across all datasets and tasks where it is currently required. Additionally, computational image restoration is becoming ever more important as imaging pipelines grow ever more complex, pushing the limits of spatiotemporal resolution in search of biological insights. Deep learning and, in particular, convolutional neural networks offer a promising path towards solutions to many of the most common problems facing the fields of BioImage analysis and restoration. However, as we attempt to adopt methods originally developed for the field of Computer Vision, we find them ill suited to some of the unique challenges of BioImage data.

In this thesis, we explore new methods for the common BioImage Analysis tasks of denoising raw microscopy data, as well as detecting, segmenting, and tracking cells and cell nuclei. In Chapter 2 we present CP-Net, a convolutional neural network designed to locate cells and nuclei by their centerpoint. We show how CP-Net can detect cells and nuclei of varying size and shape in a broad range of image types while using only very simple and widely available centerpoint annotations for training. This approach leads to state of the art detection on two datasets from the ISBI Cell Segmentation Benchmark. In Chapter 3 we track cells by combining CP-Net with a fast, parameter-free linking heuristic. We find that, in practice, detection errors are the root cause of poor tracking performance, and that high quality tracking in many cases does not require optimizing global models of cell movement. This approach to tracking leads to state of the art tracking scores on four ISBI Cell Tracking Challenge datasets. Later in Chapter 3 we investigate an extension of the tracking by assignment approach to cell tracking that incorporates a model of locally correlated motion to aid with tracking in biological tissues. In Chapter 4 we introduce STRUCTN2V, an extension the Noise2Void approach to image denoising which can deal with structured noise. We show how

structured noise is a problem common to real bioimage datasets and demonstrate a consistent improvement over standard alternatives.

While not strictly limited to fluorescence microscopy datasets of developing embryos, we validate each of our methods extensively on such data, and find that they are particularly well suited to problems with a large number of small, densely packed fluorescent nuclei across hundreds of 3D images characterized by low signal to noise ratio.

CONTENTS

1	INTRODUCTION	1
1.1	Preamble	1
1.2	A brief history of bioimage informatics	1
1.3	Current practice and problems	2
1.4	The rise of deep learning	4
1.5	Thesis preview	6
2	GENERAL PURPOSE CELL CENTERPOINT DETECTION	7
2.1	Related work	8
2.2	Motivation	11
2.2.1	Weakness of pixelwise classifiers	11
2.2.2	Comparing annotation styles	12
2.3	The CP-Net method	14
2.4	Experiments	21
2.4.1	Works on a variety of data types	21
2.4.2	Response to label noise and ambiguity	22
2.4.3	Symmetric nearest neighbors matching score	23
2.4.4	Effect of kernel shape	23
2.4.5	Effect of noise in centerpoint location	25
2.5	Detecting nuclei in 3D	26
2.5.1	Existing detection methods for embryogenesis	27
2.5.2	C. elegans nucleus detection	28
2.6	ISBI Cell Segmentation Benchmark performance	31
2.6.1	Sparse vs dense annotations	32
2.6.2	Training CP-Net for ISBI CSB	33
2.6.3	Visual Results on ISBI CSB	34
2.6.4	Scores on ISBI CSB	35
2.7	Conclusion	38
3	TRACKING CELLS AND NUCLEI	41
3.1	Related Work	42
3.2	Nearest neighbour tracking	45
3.2.1	Upper bound on linking performance	46

3.2.2	ISBI Cell Tracking Challenge	49
3.2.3	Experiment - pretrained model on Tribolium	53
3.3	Tracking with velocity gradient costs	53
3.3.1	Motivation and Related Work	54
3.3.2	Model	56
3.3.3	Experiments - twitchy Planarian	59
3.3.4	Experiments - Breast Cancer cells in dish	60
3.4	Discussion	62
4	STRUCTURED NOISE REMOVAL	63
4.1	Motivation	63
4.2	Related Work	64
4.3	StructN2V Method	66
4.4	Theory of blind spot based denoisers	70
4.5	Experiments	74
4.5.1	<i>C. majalis</i>	74
4.5.2	<i>C. elegans</i>	76
4.5.3	Ca ²⁺ imaging in mouse cortex	78
4.5.4	Nonlinear pixel response	78
4.6	Discussion	81
5	DISCUSSION	83
I	APPENDIX	87
A	APPENDIX	89
A.1	ISBI Cell Tracking Challenge dataset descriptions	91
	BIBLIOGRAPHY	99

LIST OF FIGURES

1.1	Brewer's Yeast Daguerreotype	2
1.2	The restoration and analysis pipeline	3
2.1	Pixelwise classifiers merge objects	11
2.2	Object annotation styles	12
2.3	CPNet task diagram	15
2.4	CP-Net is a U-Net	16
2.5	create training data	17
2.6	target creation details	18
2.7	code loadDataAndTrain()	19
2.8	CP-Net inference	20
2.9	CPNet detections on various data	21
2.10	Label noise corrected	22
2.11	Symmetric Nearest Neighbour Matching	24
2.12	Kernel Size experiment	25
2.13	Jitter Experiment	26
2.14	C. elegans nucleus size vs time	28
2.15	C. elegans generalization across time	29
2.16	C. elegans generalization across time	30
2.17	Collection annotation styles	32
2.18	ISBI CSB results: 2D segmentations	35
2.19	ISBI CSB results: 3D segmentations	36
2.20	ISBI CSB results: Fluo-N3DL-DRO	37
2.21	ISBI CSB results: Fluo-N3DL-TRIF	38
2.22	DET performance vs amount of training data	40
3.1	Sulston	43
3.2	Annotation number vs Performance	52
3.3	CP-Net generalization across acquisitions	54
3.4	Velocity Gradient Cost Model Diagram	56
3.5	Velocity Gradient Cost Tracking on Planarians	60
3.6	Velocity Gradient Cost Tracking on Breast Cancer Cells	61
4.1	StructN2V training	67
4.2	Mask diagram	68
4.3	Fourier analysis of noise	69

4.4	background selection	69
4.5	StructN2V masking diagram	71
4.6	StructN2V task diagram	75
4.7	StructN2V on flower dataset	76
4.8	StructN2V on C. elegans	77
4.9	Denoise Ca ²⁺ induced fluorescence images of mouse cortex	79
4.10	Removing hot pixel artifacts	80

LIST OF TABLES

2.1	CP-Net Detection on ISBI CSB	39
3.1	Tracking with ground truth centerpoints	48
3.2	CP-Net Tracking on ISBI CTC	50
4.1	Image quality metrics on flower	77
A.1	Table of ISBI Cell Tracking Challenge Parameters	90

INTRODUCTION

1.1 PREAMBLE

Bioimage informatics is a new subdiscipline of bioinformatics and computer vision which brings algorithms to bear in the service of microscopy. Two of the goals of this field are to improve the visual quality of images (*image restoration*) and to extract quantitative information about biological specimens directly from images (*image analysis*). New methods are necessary if we are to take full advantage of the massive amount of image data produced by a rapidly growing collection of microscopy techniques (Myers, 2012; Sbalzarini, 2016; Meijering et al., 2016). This thesis contributes methodological advances in three areas, image denoising, cell detection, and cell tracking, as small steps towards these goals.

1.2 A BRIEF HISTORY OF BIOIMAGE INFORMATICS

Microscopy has been an integral part of biology since the mid 17th century when Robert Hooke and Antonie van Leeuwenhoek gave the first description of cells and unicellular organisms, (Hooke, 1665). In 1844 Alfred Donné and Jean-Bernard-Leon Foucault pioneered *photomicrography*, using Foucault's improvements to the Daguerreotype system, acquiring microscopy images directly from light as opposed to creating hand-drawn diagrams while looking through an eyepiece (fig. 1.1, Batchen et al., 2004). The digitization of photomicrographs, first through scanners (Ledley, 1959; Ledley, 1964; Preston, 1976), then eventually through direct capture of digital images via Charge Coupled Semiconductor Device (CCD) (W. S. Boyle and Smith, 1970), enabled the use of computer algorithms to manipulate these pictures. Eventually, the personal computer revolution brought cheap digital storage, high resolution color screens for display, and the mouse for interaction and annotation.

New microscopy techniques were invented which rely directly on image processing algorithms (Förster et al., 2005; Betzig et al., 2006; Swoger et al., 2007; Verveer et al., 2007).

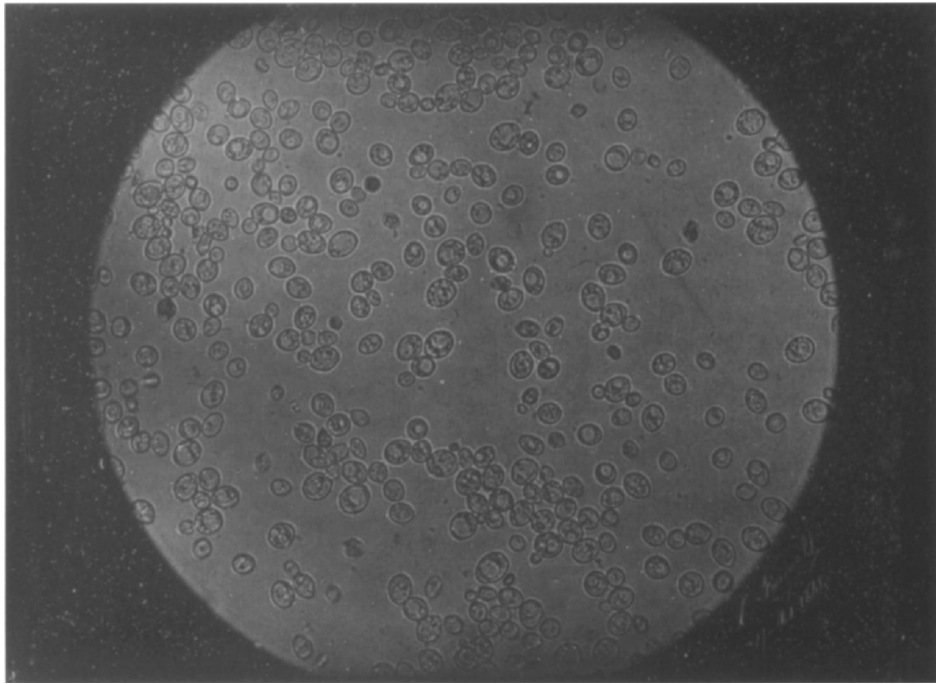


FIGURE 1.1 A Daguerreotype image of brewer's yeast acquired in 1844 by Jean-Bernard-Leon Foucault and published in Alfred Donné's *Cours de Microscopie*. (photo: The Metropolitan Museum of Art, New York. Batchen et al., 2004)

The rise of fluorescence microscopy and transgenic interventions called for image manipulation software with the ability to zoom and adjust the field of view in space and time for interactive viewing and annotation of large numbers of bioimages (Abramoff et al., 2004; Schneider et al., 2012; de Chaumont et al., 2012; Kankaanpää et al., 2012; Schindelin et al., 2012) as well as rapid sharing of simple image processing scripts.

Most recently convolutional neural networks (CNNs) have brought massive gains in accuracy across many bioimage tasks including restoration, cell detection and segmentation Weigert et al., 2018; Schmidt et al., 2018; Weigert et al., 2020; Krull et al., 2019. Yet still the field of bioimage analysis is changing faster than ever as new assays and microscopy methods are introduced and new algorithms improve the state of the art across tasks and data types in pursuit of error-free automated analyses.

1.3 CURRENT PRACTICE AND PROBLEMS

After the images are acquired biologists use algorithms to improve image quality (*bioimage restoration*) as well as to automatically extract quantitative information about the shape and movement of cells and other biological objects (*bioimage analysis*). A great diversity of algorithms have

been developed to deal with the broad spectrum of tasks, sample types and imaging modalities.

- tasks (e.g. denoising vs deconvolution in bioimage restoration, detection vs segmentation in bioimage analysis)
- sample types (e.g. membrane label in developing tissue vs label free imaging of HeLa cells in vitro)
- imaging modalities (lightsheet fluorescence microscopy vs phase contrast)

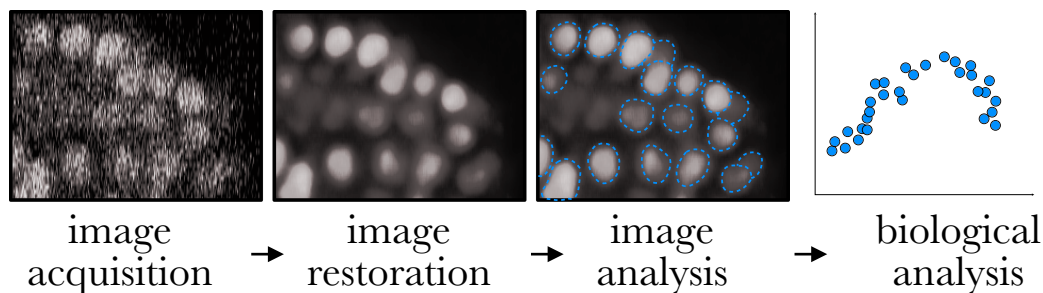


FIGURE 1.2 How do image restoration and analysis fit into the big picture of biological science? First, images are acquired by observing biological samples under a digital microscope. Then these images are restored by removing noise, blur and other artifacts. Subsequent image analysis determines the physical locations and traits of each visible object. Finally, biologists use the information gleaned from these images to form and test hypotheses.

To what extent are the problems faced by the field of bioimage informatics specific to individual tasks and datasets?

Both bioimage informatics and bioimage restoration share in their overarching, unifying goal: to help biologists extract the maximum information from the images with the minimum time and effort. Bioimage restoration algorithms achieve this by removing noise from images making it easier for our eyes to pick out the relevant parts of the signal. Bioimage analysis tools achieve this by quantifying the size, number, location and movement of cells, nuclei and other biological objects to avoid tedious manual annotation and counting by eye. The emphasis on *time and effort* highlights the fact that these methods are in competition with a variety of approaches and each has a cost.

All bioimage analysis methods are implicitly in competition with *manual analysis* (annotation by hand), which sets a high bar for accuracy but also requires a lot of time and effort. While "just do it by hand" does not apply to bioimage restoration tasks, one potential solution to problems of image quality is to re-acquire the dataset using different settings, e.g. longer exposure, a more efficient fluorescent label, a new microscope, or a new high-resolution technique. Further, both bioimage analysis and bioimage

restoration have a small set of simple baseline techniques that serve as points of comparison / competition, e.g. denoising with the median filter or detecting nuclei with local intensity max-filter or thresholding + connected components. Thus, the job of the bioimage informatician is to develop algorithms which extract information from bioimages more efficiently than purely manual analysis, re-acquiring better data, or other widely available baseline algorithms.

There are situations in which these alternatives are simply not possible.

1. We may already be using the highest quality data, or it may be that we only had one shot to acquire data, and now we can't go back to reacquire.
2. Or it could be that our dataset is several TB large and filled with objects, so it's too big to annotate by hand.
3. Or the simple restoration and analysis methods just aren't accurate enough to satisfy the statistical requirements of our experiment.

Since bioimage algorithms process images much faster than human annotation or re-acquisition the primary dimension on which they fall short is *accuracy*. A more accurate restoration algorithm reveals more of the true signal and removes more of the noise. A more accurate analysis algorithm identifies more objects correctly, reducing both false positives and false negatives. But it is sensible to assume that the time spent fixing mistakes after our analysis has run, the *curation time*, is proportional to the error rate ($1 - \text{accuracy}$). This suggests that improving the accuracy of a method from e.g. 0.94 to 0.97 may actually lead to cutting the manual curation time in half! Thus, even relatively small gains in accuracy may be significant if they correspond to significant reductions in error rate.

1.4 THE RISE OF DEEP LEARNING

In the last few years a new class of method based on deep learning and convolutional neural networks has come to provide the most accurate approach to a variety of tasks in bioimage informatics, even achieving human expert error rates on some tasks Meijering, 2020.

Cireşan was among the first to popularize Convolutional Neural Networks (CNNs) with work segmenting neural membranes in EM images Ciresan et al., 2012 and detecting mitotic nuclei in breast cancer histology slides Cireşan et al., 2013. A major breakthrough followed in 2015 when Ronneberger introduced the "U-Net" CNN architecture Ronneberger et al., 2015 with *skip connections* enabling high fidelity segmentations. Since

then bioimage methods based on CNNs have become the leading approach to some of the field's most common problems: cell segmentation in 2D and 3D Schmidt et al., 2018; Weigert et al., 2020, mitosis detection C. Li et al., 2018, and fluorescence image denoising Weigert et al., 2018.

Compared to previous machine learning approaches (Random Forests, SVMs) CNNs offer a much larger number of effectively trainable parameters, and therefore greater modeling power. This additional modeling power has unlocked large gains in accuracy, especially for tasks involving noisy imagery and complex object geometry. Further, CNN architectures are highly flexible, and require relatively few changes to the underlying model architecture when applied to different tasks, sample types or imaging modalities. Rather we only need to make relatively small adjustments to the initial or final layers of the network, swap out the training data, or change the loss function in order to adapt the U-Net model to widely divergent problems.

Despite their flexibility and accuracy, these models still fall short of human annotators when compared on the time and data required to train them on a new task. Human annotators typically require a very small number of individual examples in order to adapt to a new task, while CNNs most often are retrained from scratch, usually requiring several hours of GPU time. Even seemingly small changes to the data domain can significantly reduce the performance of CNN models. While work on zero-shot and one-shot learning exists it has yet to make a large impact in bioimage analysis. Still, the *data efficiency* problem is significant in bioimage informatics.

Unfortunately, we don't yet know how to create highly generic models capable of high performance across tasks, sample types and image modalities in bioimage informatics. Neither have we created models capable of one- or few-shot learning new bioimage analysis tasks. Given the absence of these model classes from the literature *self-supervised methods* are an appealing alternative. Self-supervision refers to the ability of a model to train itself using only the raw input data. Training times may be long, but at least it doesn't require expensive human effort like manually labeling images or acquiring multiple datasets from the microscope. However self-supervised approaches are unable to offer state of the art performance on certain tasks like cell detection. For these tasks it makes sense to search for *data efficient* supervised techniques that minimize the amount of effort required to create training data.

1.5 THESIS PREVIEW

The tradeoffs listed above motivate our work on self-supervised and data efficient convolutional neural networks for image denoising and cell detection.

Chapter 2 introduces CP-Net , a model of *cell detection* based on CNNs for pixelwise regression which relies only on centerpoint dot annotations rather than full instance annotations for training. We show how this model extends similar approaches used in the histology and mitosis detection literature, and test it’s performance across a broad range of datasets by participating in the ISBI Cell Segmentation Challenge, achieving broadly competitive results and multiple positions on the leader board.

Chapter 3 extends CP-Net with a greedy, parameter free discrete linking algorithm which enables tracking multiple dividing cells over time. While deep learning has become the dominant approach for multiple problems in bioimage informatics *cell tracking* is most often treated as a discrete data-association problem relying on combinatorial optimization techniques and traditional solvers. We submit this model to the ISBI Cell Tracking Challenge and achieve multiple 1st place scores on datasets of developing embryos, demonstrating that a very simple discrete linking algorithm is sufficient for state of the art performance if we begin with high quality detections from a CNN.

Later, Chapter 3 develops a new graphical model for tracking when our parameter-free tracker fails on data with large cell displacements. This model introduces “velocity gradient costs” to the Tracking By Assignment framework to account for locally correlated motion between neighboring cells and is solveable with standard techniques from combinatorial optimization.

Finally Chapter 4 introduces STRUCTN2V, a self-supervised image denoising model which can handle creating kinds of *locally correlated noise*. Our approach is an extension of the Noise2Void method and likewise is independent of network architecture. We demonstrate and flexible enough to deal with 2D and 3D correlated noise as it appears in real datasets even including hot pixels.

GENERAL PURPOSE CELL CENTERPOINT DETECTION

Detecting cells and nuclei is the critical first task in many BioImage Analysis workflows, allowing us to infer the number of objects and their spatial locations. Successful approaches are often highly data-specific, and do not necessarily generalize well across the large and growing number specimens, labels, and types of microscopy. This is especially true for 3-D+T images of developing organisms, where the large data size, high object count and poor SNR have historically required specialized solutions for cell detection and tracking. Indeed, in 2014, an investigation of the most common heuristic approaches for 3-D cell detection in a variety of common microscopy modalities (Schmitz et al., 2014) determined that “the results demonstrate that the current automated methods suffer from lower detection rates and higher false-positive rates than are acceptable for obtaining valid estimates of cell numbers”, leaving users with no choice but to perform arduous manual cell counting, potentially even according to non-intuitive “stereologic” counting rules in very large samples.

In this chapter we introduce CP-Net , our extension of the *structured regression* approach to cell detection. Our approach predicts cell centroids across a wide variety of BioImage datasets, including anisotropic 3-D images, and can be trained using only widely available centerpoint dot annotations. We investigate this model in detail, explore its robustness to label noise, and offer techniques for dealing with sparse and unbalanced datasets. We demonstrate its generality by detecting cells and nuclei across a broad spectrum of datasets via participation in the ISBI Cell Segmentation Benchmark. We show that CP-Net has good broad spectrum performance with minimal hyperparameter tuning and rapid inference. In some cases CP-Net is competitive with or exceeds the state of the art.

Contributions

- Introduce CP-Net , a variant of the *structured regression* approach to object detection based on U-Net which has been designed to work with 2D and anisotropic 3D BioImage data.
- Present results from the ISBI Cell Segmentation Benchmark where CP-Net takes 1st place in object detection in two categories, as well 1st place in the sum total detection score across all datasets.
- Show how structured regression can be adapted for sparse annotations and anisotropic 3D data.
- Study CP-Net robustness to human error in centerpoint label location as well as with respect to hyperparameters for training data construction.
- Study CP-Net data efficiency and show benefits of carefully selecting a diverse training set.

2.1 RELATED WORK

While automated cell detection is a task that has interested researches for decades, the increase in speed, accuracy and breadth of applicability of these techniques in recent years has been stunning. This is due to a number of factors, but can be summarized simply as “the rise of deep learning”. In this section we study the history of advances in deep learning for centerpoint detection in 3D images of densely packed cells, with a specific focus on models trainable via *weak annotations*, i.e. requiring only cell centerpoints or bounding boxes as ground truth.

Although 3-D data is quite common in the field of Medical Image Analysis (MIA), the most common tasks are semantic segmentation of organs and tissues, not detection of densely packed object instances. However, detection and classification of nuclei in 2-D histology slides is a critical task in cancer screening, and the MIA community has made great strides towards fully automated approaches, largely through the use of deep neural networks. Sjöström et al., 1999 were first to use an artificial neural network (ANN) to detect cells. They used a 3-layer artificial neural network (ANN) with over 40 total parameters, they broke the image up into small patches which were manually classified as either “containing a cell” or not. The classes were assigned numerical values of 0.9 and -0.9 and predicted by a final tanh activation layer to minimize a maximum-absolute-error loss. Prediction was performed using a sliding-window over an image of arbitrary size. Ning et al., 2005 train a convolutional neural network (CNN) with four convolutional and two pooling layers

to classify 40×40 image patches of 2D differential interference contrast images of early stage *C. elegans* embryos, enabling downstream nucleus detection. However the output of the CNN was too noisy to allow direct detection, and required a markov random field to denoise the predicted labels before detection could occur. Arteta et al., 2012 generate a superset of cell proposals using Maximally Stable Extremal Regions (MSER) (Matas et al., 2004) which are filtered and classified by a support vector machine (Boser et al., 1992) trained on Ground Truth image regions which have been labeled with a single centerpoint dot annotation.

Much of the progress can be attributed to highly generic improvements in network architecture and training over task-specific innovations. Faster machines and GPUs have enabled the use of neural networks with more layers (*deeper*) and more neurons per layer (*wider*), dramatically improving accuracy, (Uetz and Behnke, 2009; Strigl et al., 2010; Cireşan et al., 2012; Krizhevsky et al., 2012). Cireşan et al., 2013 introduce *max-pooling* layers along with standard convolutional and fully connected layers in a CNN with over 13 *thousand* parameters to detect mitotic nuclei in histology slides, obtaining breakthrough results on the ICPR 2012 mitosis detection competition. Prediction was done in a sliding window manner, but speeds were greatly improved over prior CNN implementations by making use of graphics processing unit (GPU) hardware for forward and backward passes through the net. Long et al., 2015 propose the “fully convolutional” (FCN) architecture which enables the simultaneous prediction of an entire image patch (instead of a single pixel) with a single forward pass of the network. Ronneberger et al., 2015 propose an alternative fully-convolutional architecture which features two mirror-symmetric paths composed of an equal number of down- and up-sampling layers attached via “skip connections” which combine spatially precise low-level features with deeper, more semantic features. He et al., 2016 simplify the training of deeper networks by using residual connections to help to avoid the “vanishing gradient problem”.

Bounding box detection is common in Computer Vision applications where such annotations are widely used, but has also seen some use lately in BioImage Analysis. Girshick et al., 2014; Girshick, 2015 introduce the “R-CNN” family of models using a CNN-based approach similar to Arteta et al., 2012 to classify objects from a large pool of bounding-box proposals created using heuristic methods like Selective Search (Uijlings et al., 2013). Ren et al., 2015 extended the R-CNN framework with a convolutional Region Proposal Network (RPN) which replaces Selective Search with a CNN which learns to propose bounding box candidates directly from raw pixels. Akram et al., 2016a; Akram et al., 2016b; Akram et al., 2017 extend RPNs to generate cell proposals for various 2-D fluorescence and phase contrast microscopy datasets enabling better cell

segmentation and tracking. Redmon et al., 2016 You Only Look Once (YOLO) combined the proposal generation and filtering tasks into a single network for simultaneous bounding box detection and classification in a single forward pass through the network. Avoiding the one-pass-per-box filtering step enabled fast, real-time frame rates. Zhao et al., 2018 extend Region Proposal Nets to obtain 3-D bounding boxes which are coupled with a binary pixelwise classifier to obtain instance boundaries similar to Mask R-CNN (He et al., 2017).

Centerpoint annotation based methods have also progressed as new network architectures, tasks and losses have been introduced. Cireřan et al., 2013 use a patch-to-pixel architecture to perform pixelwise classification. Y. Xie et al., 2015 use a “deep voting” approach in which a network predicts 2-D centerpoints explicitly instead of an image with pixel classes.

This approach was adapted by Sirinukunwattana et al., 2016 who use an explicit representation of points internally before evaluating the network loss on an image representation. Such formulations lose the translation invariance inherent to fully convolutional networks, but are able to encode cell density constraints into the architecture. Hirsch and Kainmueller, 2020 show that a form of centerpoint voting is a useful auxiliary task when training networks for 3-D segmentation. Sirinukunwattana et al., 2016 include an explicit representation of centerpoints inside the network, while still predicting an image in the final layer which is used for pixelwise regression and training. Kainz et al., 2015 introduce detection by “structured regression” and suggest a target kernel which decays exponentially with distance to the nearest centroid, but use 2D Random Forests instead of ConvNets. Y. Xie et al., 2018 extend the structured regression approach, adopting the exponential target kernel of Kainz et al., 2015, but greatly improve results by replacing random forests with a 2-D residual U-Net. Höfener et al., 2018 evaluate structured regression models with multiple target kernels on 2-D histology data including pixelwise classification of square blocks, $1/x$ style kernels, and Gaussian.

Centerpoint pixel classification has proved useful in 3-D fluorescence data. Thierbach et al., 2018 use a centerpoint pixel classification approach in 3D as the seeds for instance segmentation. Recently, Sugawara et al., 2021 use a 3D U-net and a “doughnut-style” target kernel to classify pixels into one of three classes depending on their distance from nuclear centerpoints. And some work has been done to extend the “structured regression” approach to detection to 3-D data. Waithe et al., 2016 extend the density estimation approach to 3D for the first time, using Random Forests and custom low-level image features.

In recent work Gomariz et al., 2021 use a 3D Bayesian CNN to regress Gaussian kernels placed at mesenchymal stromal cell centers for de-

tection with properly calibrated uncertainty information. Finally, Malin-Mayor et al., 2021 also adopt a very similar approach to ours for cell detection, and further complementing it with an additional network trained on tracking ground truth to predict cell motion in 3D.

Most detection methods, including bounding-box and centerpoint regression models listed above, include a non-differentiable non-maximum suppression (NMS) post-processing step which must be calibrated during training. End-to-end object detection methods which directly propose conflict-free sets of detection candidates are an area of ongoing research (Stewart et al., 2016; Carion et al., 2020), but as of yet lag in performance vs NMS-based alternatives.

2.2 MOTIVATION

2.2.1 WEAKNESS OF PIXELWISE CLASSIFIERS

Probably the most common learning based approach to cell detection is instance segmentation based on pixelwise classifiers. This approach classifies every individual pixel, e.g. as “foreground” or “background” and then to groups the foreground pixels into connected components representing cells. A common implementation of this approach is Ilastik, Berg et al., 2019; Sommer et al., 2011. However, such approaches are sensitive to misclassified pixels on object boundaries, which often lead to false-merges. Such problems are most extreme when objects are densely packed together (see fig. 2.1), when the signal-to-noise ratio (SNR) of the image is low, or when the objects have large surface area.

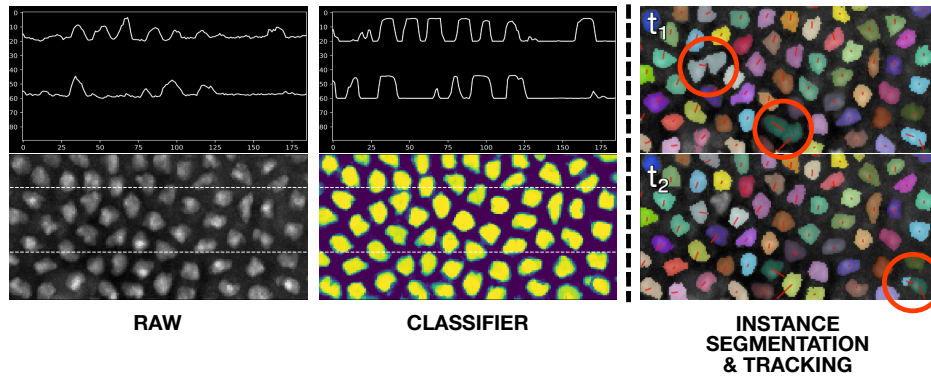


FIGURE 2.1 Approaches that begin with pixelwise classification tend to merge touching objects. Red circles show false merges/splits from an instance segmentation obtained via thresholding the pixelwise classification (middle). Note also how these segmentation errors lead to downstream tracking errors (red line segments show cell movement). RAW shows 2D slice from developing *D. melanogaster* nuclei with fluorescent Histone2B protein (data from Keller Lab, Janelia Farms via celltrackingchallenge.net).

It is possible to compensate for some of the problems of pixelwise classifiers with additional downstream processing. For example Schiegg et al., 2013 perform joint cell segmentation and tracking, which follows objects in undersegmented *clusters* and enforce constraints on the number of cells in a cluster that accounts for splits, merges and cell divisions. This clever approach works well if timeseries data is available, objects mostly remain in the field of view and the object count is relatively stable, but can be computationally demanding. Alternatively, powerful denoising can be applied to make object boundaries less ambiguous, which we have shown improves downstream pixelwise classification and segmentation (Weigert et al., 2018). Pixelwise classifiers have also been used in combination with sophisticated object shape models (Richmond et al., 2015) and/or progressive merging via heuristic shape dependent merge rules (Blasse, 2016), (Stapel et al., 2016) in order to minimize false merges. These approaches have limited generality across types of BioImage data and require complex, custom training procedures. However, pixelwise classifier based methods remain popular because of their flexibility, easy training, and because their reliance on simple “scribble annotations”, see fig. 2.2, makes annotation simple.

2.2.2 COMPARING ANNOTATION STYLES

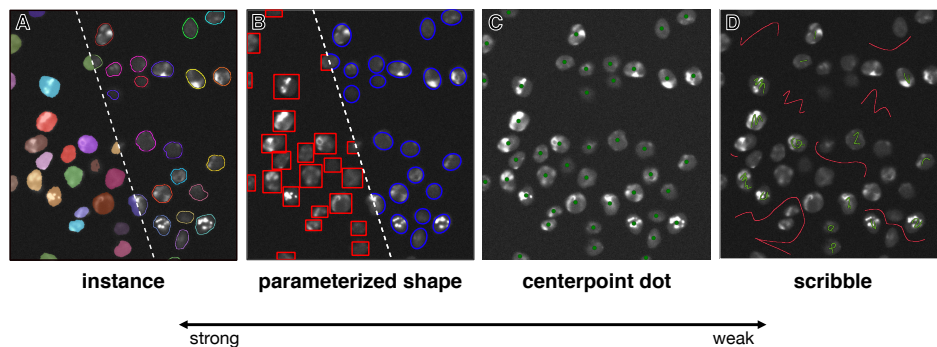


FIGURE 2.2 Object annotations differ greatly in the amount of information they provide about the location and geometry of objects in the image. Here place a small set of the most common annotation types onto a spectrum based on their “strength”, i.e. the amount of information they encode about object shape. **(A)** *Instance annotations* describe the exact (rasterized) shape of objects. Objects are associated with unique integer labels. **(B)** *Parameterized shape annotations* e.g. ellipsoids and bounding boxes. Allows us to use continuous valued parameters for sub-pixel precision. But usually is an approximation of the full instance shape. **(C)** *Centerpoint dot annotations* consist of just a single point located at the “center” of each object. Don’t contain any information about object shape! **(D)** *Scribble annotations* don’t contain any information about individual object instances at all. They merely group sets of pixels into classes.

Image annotations come in a wide variety of styles, from whole image descriptors and tags used in image classification tasks, to extremely de-

tailed instance segmentation pixelwise labels. In Computer Vision bounding box annotations are a popular way to mark the approximate size and location of objects. For manual cell counting tasks it is common to mark each cell with a single point near the center. In fig. 2.2 we place a few of these annotation styles on a spectrum with “strong” *instance annotations* occupying the position furthest left while “weak” *scribble annotations* on the right are the least informative, but the simplest to create.

Scribble annotations in fact tell us nothing about object *instances* in the scene, instead being merely class labels for individual pixels. Despite this they are often used to train pixelwise classifiers based on Random Forests (Breiman, 2001) or U-Nets (Ronneberger et al., 2015) which are a common intermediate model on the way to generating a semantic or instance segmentation. Commonly used tools to visualize and create such annotations include Labkit (Arzt et al., 2022) and Ilastik (Sommer et al., 2011).

Centerpoint dot annotations require just a single point annotation at the center of an object. They are commonly used to count objects and to track individual cells over time, however, they tell us nothing about the size and shape of objects. Common software for manual annotation includes the multipoint annotation tool in Fiji (Schindelin et al., 2012), the Points Layer in Napari (Sofroniew et al., 2020), or H.S. Peng’s specialized V3D tool which under certain circumstances enables annotating points in 3D with a single click (Peng et al., 2010). Centerpoint annotations have the benefit of only minimally obscuring the underlying data when visualized, which is important in densely packed scenes. It is common to use local max filters to infer approximate object centerpoints in fluorescence images.

Parameterized shape annotations tell us about object size and location and provide some amount of shape information, for example using *bounding boxes* or *ellipsoids*. Software tools like AceTree (T. J. Boyle et al., 2006), CATMAID (Saalfeld et al., 2009) and Mammot (Wolff et al., 2018) make it possible to visualize and edit spheres and ellipse with a combined rendering of image data and annotations. The StarryNite (Bao et al., 2006) method detects cell nuclei using a spherical model. Faster R-CNN’s Region Proposal Networks (Ren et al., 2015) and You Only Look Once (YOLO) (Redmon et al., 2016) are CNN approaches that return 2D bounding boxes around objects in natural images. Tracking By Gaussian Mixture Models (TGMM) (Amat et al., 2014) fits ellipsoids to fluorescent nuclei while the Generalized Hough Transform (Ballard, 1981) and the CNN based While StarDist models (Schmidt et al., 2018; Weigert et al., 2020) predict complex shapes with a large number of parameters.

Full *object instance annotations* provide the entire rasterized shape of an object, usually encoded as images where pixel have integer values corresponding to unique object instance ID's (not intensity values). We can manually draw such labels with specialized image viewing and annotation tool such as a Napari Labels Layer (Sofroniew et al., 2020). Note, this requires marking each pixel contained in each cell. Variations exist on this encoding which permit overlapping objects. Common instance segmentation methods include the traditional heuristic Watershed method (Beucher and Lantuéjoul, 1979), and modern deep learning based approaches like Mask R-CNN (He et al., 2017).

Not every type of annotation is appropriate for any type of object. A spherical parameterized shape model is obviously inappropriate for filaments. Centerpoint annotations might be inappropriate for very large objects without an obvious center or for long filaments. Instance segmentation annotations are likely not a good fit for objects without well defined boundaries, e.g. cells that blend continuously into the background or clouds in natural images. Note that both centerpoint dots and parameterized shapes may actually be a more precise description of object location or geometry than is possible with pixel-based labels as they are not subjected to rasterization on the pixel grid. This is exploited in single molecule localization techniques used in superresolution microscopy like PALM (Betzig et al., 2006), STORM (Rust et al., 2006) and DECODE (Speiser et al., 2020).

In general the different annotation types represent a tradeoff between the amount of work required to make them and how informative they are about the scene. This tradeoff is relevant both for biologists who want to extract knowledge from the scene using the annotations directly, as well as for BioImage analysts who aim to use the annotations to optimize cell detection and segmentation models. Despite this tradeoff not all annotations types are created equal, and centerpoint dot annotations in particular offer a nice balance of easy to create, semantically meaningful, and useful for training further analysis algorithms.

2.3 THE cp-net METHOD

CP-Net is a general purpose, learned method to detect cells and nuclei based on centerpoint annotations. CP-Net can detect objects across a broad spectrum of biological image data, including densely packed 3D tissues, and minimizes false merges that plague approaches based on pixel classifiers.

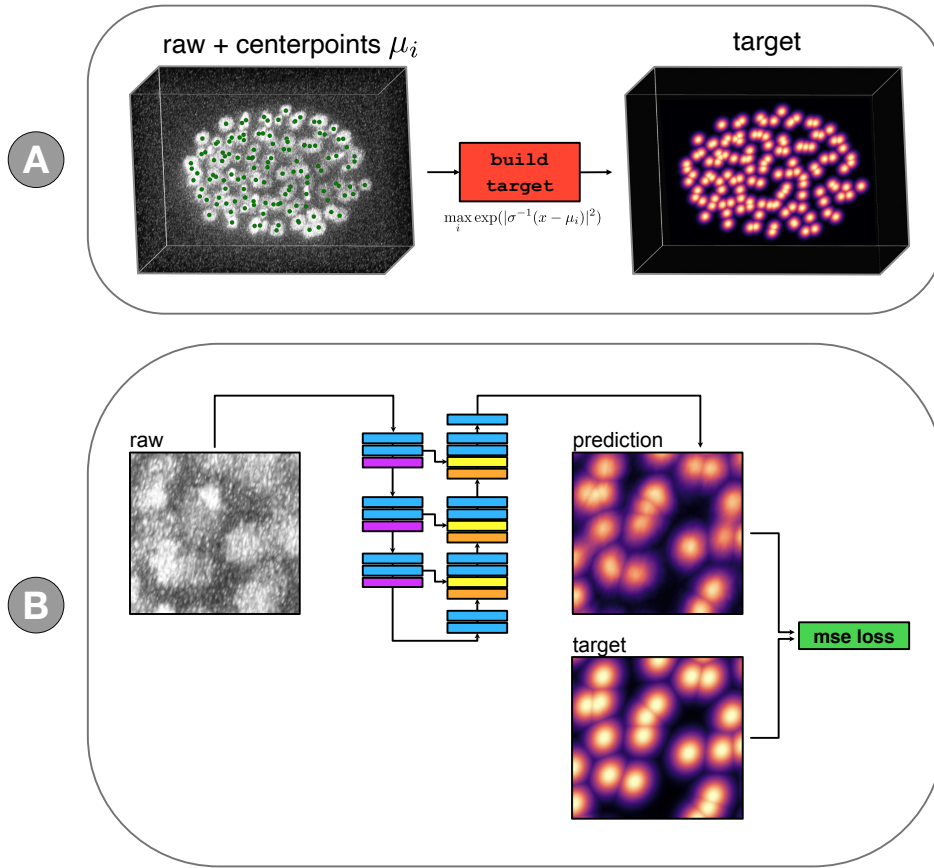


FIGURE 2.3 (A) Target images are generated from a combination of raw stacks and manually created centerpoint annotations according to `createTarget()`. We extract centerpoints from the predicted image by finding local intensity maxima with `findPeaks()`. (B) CP-Net is trained on a sequence of image patches to regress target given raw input. The mean squared error (MSE) is evaluated per-pixel and the gradient of that error is backpropagated through the network to update the weights in each convolutional layer (blue).

At its core the CP-Net method is based on an image-to-image convolutional neural network (CNN) which converts each raw image into a smooth, “structured regression” image where each cell or nucleus is transformed into a bright blob with a peak at its center. Centerpoints are extracted from these predicted images by finding local intensity peaks.

In this section we show a comprehensive diagram of the three MaxPool U-Net architecture we used for all subsequent experiments. We also provide annotated python code detailing the three main parts of the CP-Net method: data generation, training, and prediction. Additionally, we provide python code on the generation of the target images from centerpoint annotations.

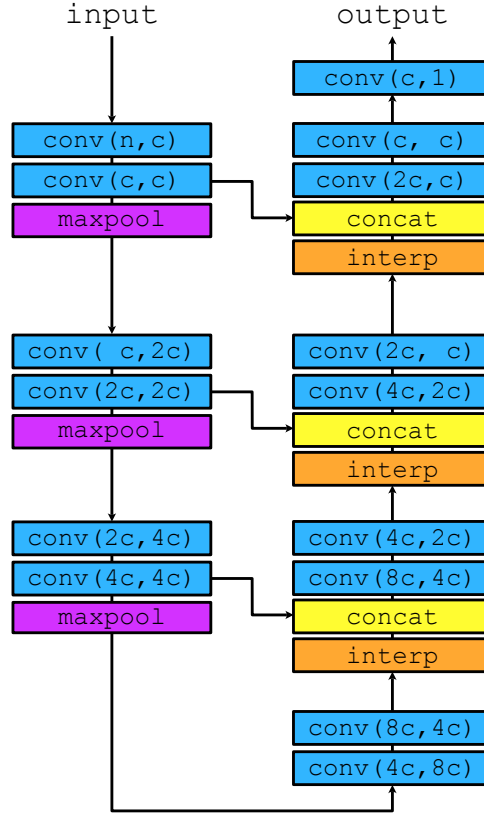


FIGURE 2.4 The architecture for CP-Net is a standard U-Net (Ronneberger et al., 2015; Çiçek et al., 2016) with three max-pooling layers on the downward path, and convolutional kernels with size 5×5 and $3 \times 5 \times 5$ for 2D and 3D data respectively. Our U-Net accept images, input, with any number channels (n), although all ISBI datasets have $n=1$. Convolution blocks, $\text{conv}(a, b)$, are fully connected across input and output feature dimensions, a, b . These blocks are parameterized by the number of features chosen in the first layer, c , which we set to 16 for all experiments. Pooling layers, maxpool , uses a 2×2 window in X, Y dimensions only. Interpolation blocks, interp , perform a 2×2 nearest-neighbour upsampling in X, Y dimension only. Tensor concatenation, concat , is performed across the feature dimension. The result, output, has the same spatial shape as the input, with one channel and one batch. The *theoretical receptive field* is 184px (29px) in XY (Z) while the full-width half-max of the pre-training *effective receptive field* (Luo et al., 2016) is 51px (7px) in XY (Z).

```

def createTrainingDataSingle(raw, labeled_image, scale):

    # Extract centerpoints from integer labeled ground truth images. Be careful
    # NOT to scale labeled_image before extracting centerpoints because thin
    # labels may disappear!
    gt_pts = extractPtsFromLabels(labeled_image)

    # Linearly interpolate (downsample) raw image and centerpoints spatial
    # dimensions(anisotropically) so objects fit in CP-Net effective receptive
    # field (approx 51px XY, 7px Z).
    raw_scaled = zoom(raw, scale)
    gt_pts_scaled = floor((gt_pts+0.5)*scale-0.5)

    # Normalize raw intensity via per-image affine rescaling histogram
    # percentiles (2, 99.4) -> pixel values (0, 1). NOTE: No clipping. Allow
    # for pixel values outside [0,1] as input to CP-Net.
    raw_scaled_normalized = normalizePercentileTo01(raw_scaled)

    # Generate target with same shape as raw_scaled
    # sigmas = (5,5) in 2D, (3,5,5) in 3D for almost all ISBI datasets
    target_shape = raw_scaled.shape
    target = createTarget(gt_pts_scaled, target_shape, sigmas)

    # Break raw and target into non-overlapping patches small enough that CP-Net
    # fits in 8GB GPU memory. Patch size (256,256) in 2D, (16,128,128) in 3D.
    # If annotations are INCOMPLETE, throw away patches without any
    # centerpoints.
    patches = patchesForTraining(raw_scaled_normalized, target)

    # Regenerate gt_pts for each patch to avoid border clipping problems when
    # computing fl_snn validation score. If annotations are SPARSE, edit `mask`
    # to only show target pixels near known annotations.
    for p in patches:
        p.gt_pts = findPeaks(p.target, nms_shape)
        if SPARSE:
            p.mask = (p.target > exp(-0.5*(3**2))) ## cutoff at 3 sigma
        else:
            p.mask = ones(p.target.shape)

    savePatchesToDisk(patches)

def findPeaks(img, nms_shape):
    plm = skimage.feature.peak_local_max
    return plm(img, threshold_abs=0.5, footprint=ones(nms_shape))

```

FIGURE 2.5 Python pseudocode showing overview of CP-Net training data creation for a single labeled_image. To create full train set we pool across all saved patches and randomly assign them to train/vali sets with a 7/2 split. scale is the primary parameter which needs to be adjusted to make cells and nuclei fit inside the receptive field while nms_shape and gaussian_sigmas are constant across almost all datasets. We downscale images so that the objects we wish to detect fit easily into CP-Net's effective receptive field with a sufficiently large window that the object's centerpoint can be uniquely identified. Rescaling images is done with `scipy.ndimage.zoom(img, scale)` using linear interpolation (Virtanen et al., 2020). We don't pre-filter images before rescaling and find in practice this saves time and does not cause visual aliasing. Properly scaling centerpoints to match the image scaling must be done carefully to avoid off-by-one errors. A centerpoint with coordinate 0 in some dimension typically corresponds to a continuous coordinate at the center of the first pixel, i.e. at a coordinate of $1/2$.

```

def createTarget(pts, target_shape, sigmas):
    s = array(sigmas)
    ks = floor(7*s).astype(int)    ## extend support to 7/2 sigma in every direc
    ks = ks - ks%2 + 1             ## enforce ODD shape so kernel is centered!

    ## create a single Gaussian kernel array
    def f(x):
        x = x - (ks-1)/2
        return exp(-(x*x/s/s).sum()/2)
    kern = array([f(x) for x in indices(ks)].reshape((len(ks),-1)).T).reshape(ks)

    target = zeros(ks + target_shape) ## include border padding
    w = ks//2                         ## center coordinate of kernel
    pts_offset = pts + w              ## offset by padding

    for p in pts_offset:
        target_slice = tuple(slice(a,b+1) for a,b in zip(p-w,p+w))
        target[target_slice] = maximum(target[target_slice], kern)

    remove_pad = tuple(slice(a,a+b) for a,b in zip(w,target_shape))
    target = target[remove_pad]

    return target

```

FIGURE 2.6 Python code showing how to create target quickly and precisely, avoiding slow python loops and off-by-one-pixel errors. CP-Net is trained to regress a target image of uniformly sized gaussian kernels centered on each ground truth object. This target is the same size as the raw input image, and takes on the value zero everywhere except for in the vicinity of each centerpoint, where we place gaussian kernels. Target is described by

$$\text{target}(x) = \max_i \exp\left(\frac{1}{2}|\Sigma^{-1}(x - \text{pts}[i])|^2\right)$$

where Σ is a diagonal matrix that allows our gaussian kernel to adapt to the anisotropic voxel dimensions. This equality ignores the effects of finite domain / support for gaussian kernels, which squishes values 2.19×10^{-3} at 3.5 sigma or greater to zero.


```

def loadDataAndTrain():
    patches = loadPatchesFromDisk()
    if CONTINUE_TRAINING: cpnet.loadWeightsFromDisk()

    # initial values for validation metrics
    vali_best = SN(mse=inf, height=0, f1=0) # mse, peak height, f1

    for i_epoch in range(300):
        shuffle(patches.train)
        for p in patches.train:

            # Augment raw, target, mask and gt_pts with flips across XYZ and
            # 90deg XY rotations. Further augment raw with random affine
            # transform on intensity with scale in (0.8, 1.2) and shift in
            # (-0.1, 0.1).
            p = augmentRandom(p)

            # Put augmented raw patch through CP-Net.
            pred_patch = cpnet(p.raw)

            # Backpropagate mean squared error loss and update model weights
            # with Adam optimizer.
            mse = sum(p.mask*(pred_patch-p.target)**2)
            mse.backward()
            cpnet.step()

        ## Compute validation metrics across vali set and save best weights.
        vt = array([0, 0, 0, 0, 0]) # "Validation Totals"

        for p in patches.vali:
            pred_patch = cpnet(p.raw)
            pts_patch = findPeaks(pred_patch, nms_shape)

            mse = sum(p.mask*(pred_patch-p.target)**2)
            height = max(pred_patch)
            # symmetric nearest neighbor matching
            snn = snnMatching(pred=pts_patch, gt=p.gt_pts)

            vt += array([mse, height, snn.TP, snn.T, snn.P])

        N = len(patches.vali)
        # compute f1 score on combined detections across patches
        vali_epoch = SN(mse=vt[0]/N, height=vt[1]/N, f1=2*vt[2]/(vt[3]+vt[4]))

        ve = vali_epoch; vb = vali_best;
        if ve.mse < vb.mse:
            vb.mse = ve.mse; save(cpnet.weights, 'weights_mse')
        if ve.height > vb.height:
            vb.height = ve.height; save(cpnet.weights, 'weights_height')
        if ve.f1 > vb.f1:
            vb.f1 = ve.f1; save(cpnet.weights, 'weights_f1_snn')

```

FIGURE 2.7 Python pseudocode detailing how models are trained and loss and validation scores are computed, including symmetric nearest neighbour matching fig. 2.11. We use pytorch to construct the model and perform backpropagation and Adam for optimization (Kingma and Ba, 2015). We use Augmend to perform simple augmentations including rotations, flips, and affine intensity scaling. We use scikit-image to findPeaks(), (van der Walt et al., 2014).

```

def predict():
    # Linearly interpolate (downsample) raw spatial dimensions
    # (anisotropically) until objects fit in CP-Net receptive field.
    raw_scaled = zoom(raw, scale)

    # Normalize raw intensity via per-image affine rescaling. Histogram
    # percentiles (2, 99.4) -> pixel values (0, 1). NOTE: No clipping. Allow
    # for pixel values outside [0,1] as input to CP-Net.
    raw_scaled_normalized = normalizePercentileTo01(raw_scaled)
    scaled_shape = raw_scaled_normalized.shape

    # Like training, prediction is also performed patchwise because of GPU
    # memory constraints, but the patches can be larger because we don't need
    # to save tensors at each layer for the backward pass. Raw image volumes
    # are tiled into overlapping `patch.outer` slices, and each `patch.outer`
    # tile has a `patch.inner` region which is free from border effects. Patch
    # sizes are divisible by 8 in XY dimensions to match the discrete
    # translational symmetry of CP-Net with 3 MaxPool layers. Patches are not
    # reified, instead, to save memory, we return slices which produce patches
    # via `img[slice]`
    #
    # If 3D: patch_shape = (32,400,400); prediction_border = (6,40,40).
    # If 2D: patch_shape = (600,600); prediction_border = (32,32).
    patch_slices = patchSlicesForPrediction(scaled_shape)

    # Apply CP-Net to each patch, remove the region with border effects,
    # and paste the result into container.
    container = zeroArray(scaled_shape)
    for p in patch_slices:
        container[p.container] = cpnet(raw_scaled_normalized[p.outer])[p.inner]

    # Local peak detection with small, rectangular, non-max suppression with
    # nms_shape is [3,]5,5 in [Z]YX. Reject points with peak < 0.2.
    centerpoints_scaled = findPeaks(container, nms_shape)

    # Invert downscaling to get back to original image size.
    centerpoints = floor((gt_pts+0.5)/scale-0.5)

    # ISBI Challenge requires removal of objects near image borders.
    # `isbi_border` is dataset specific with value 0px, 25px, or 50px.
    if FILTER_BORDERS:
        centerpoints = filterBorders(centerpoints, raw.shape, isbi_border)

    # ISBI Challenge requires a labeled image for DET/SEG evaluation.
    # `radius` describes sphere (in real space) which is discretized
    # on (anisotropic) voxel grid to make segment labels.
    if SEGMENT:
        seg_img = placeHardSphereLabels(centerpoints, raw.shape, radius)

    # ISBI Challenge requires removal of unannotated objects for three sparse
    # datasets: Fluo-N3DL-DR0, Fluo-N3DL-TRIC, Fluo-N3DL-TRIF
    if SPARSE:
        seg_img = filterOnGtAtT0(seg_img, labels_gt_t0)

```

FIGURE 2.8 Python pseudocode showing how CP-Net inference, transforming raw into ISBI-CTC-appropriate centerpoint predictions and segmentations.

2.4 EXPERIMENTS

2.4.1 WORKS ON A VARIETY OF DATA TYPES

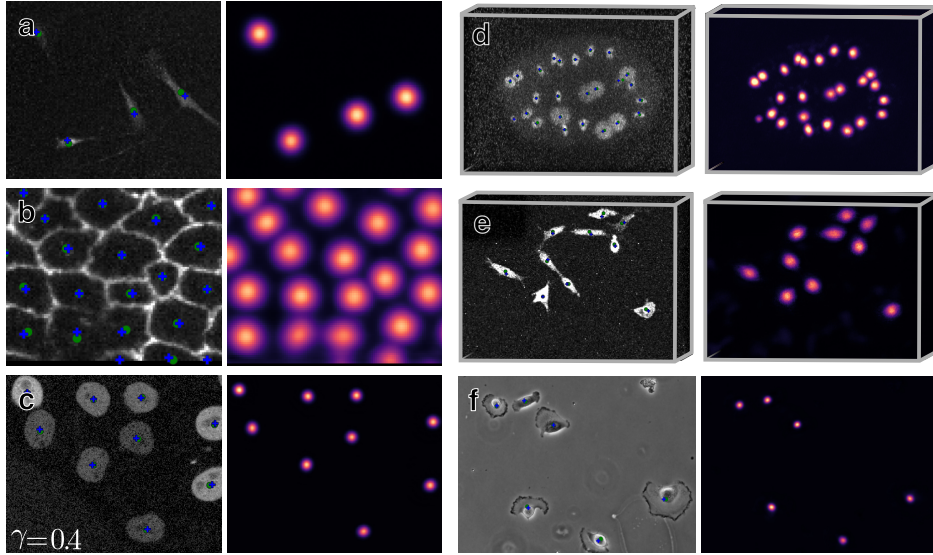


FIGURE 2.9 CP-Net can detect cells and nuclei across a broad spectrum of microscopy data. For each panel pair left shows raw image with ground truth (green) and predicted (blue) centerpoints. Right shows the output of CP-Net. Final centerpoints are local maximum of CP-Net prediction. Datasets are from the ISBI Cell Tracking Challenge (Maška et al., 2014; Ulman et al., 2017). (a) F1uo-C2DL-MSC Rat mesenchymal stem cells with fluorescent cytoplasmic label. (b) Developing *D. melanogaster* wing with e-cadherin GFP. (c) F1uo-N2DH-GOWT1 GFP-GOWT1 mouse stem cells. (d) F1uo-N3DH-CE *C. elegans* early embryogenesis with GFP-Histone. (e) F1uo-C3DL-MDA231 MDA231 human breast carcinoma cells infected with a pMSCV vector including the GFP sequence. (f) PhC-C2DH-U373 Glioblastoma-astrocytoma U373 cells under Phase Contrast microscopy.

In fig. 2.9 we show a sample of cell and nucleus detection applications on a variety of common types of microscopy data. Note how CP-Net is able to detect cells with both round and elongated shapes, with nuclear, cytoplasmic, or membrane based markers in brightfield, phase contrast, and fluorescence microscopy. Panel (a) shows elongated Rat mesenchymal stem cells with a fluorescent cytoplasmic label. Despite the elongated shape CP-Net is able to provide a reasonable estimate of the location of the manually placed annotations. Panel (b) shows Ecadherin-GFP labeled cell membranes from *D. melanogaster* wing epithelium. CP-Net finds centers of cells with varying size and provides good estimates when boundaries are partially missing. Panel (c) shows GFP-GOWT1 labeled mouse stem cells. CP-Net is able to detect these cells despite the high relative background noise levels and the extreme differences in cell expression levels. Note the lower $\gamma=0.4$ to bring the otherwise invisible darker cells out from the background. Panel (d) shows the developing *C. elegans* early embryo at the 24 cell stage. The GFP-Histone label re-

veals cells in different phases of the cell cycle and we see that CP-Net is able to detect nuclei with geometry and appearance associated with each phase despite low resolution and high background noise levels. Panel (e) shows MDA231 human breast carcinoma cells infected with a pMSCV-GFP vector. Here CP-Net is able to find the center of oblong shapes in 3D in heavily oversaturated images. Panel (f) shows phase contrast images of Glioblastoma-astrocytoma U373 cells. CP-Net is able to find cell centers despite variety of textures and to distinguish cells from round background artifacts of similar size and dead cell material (top right).

2.4.2 RESPONSE TO LABEL NOISE AND AMBIGUITY

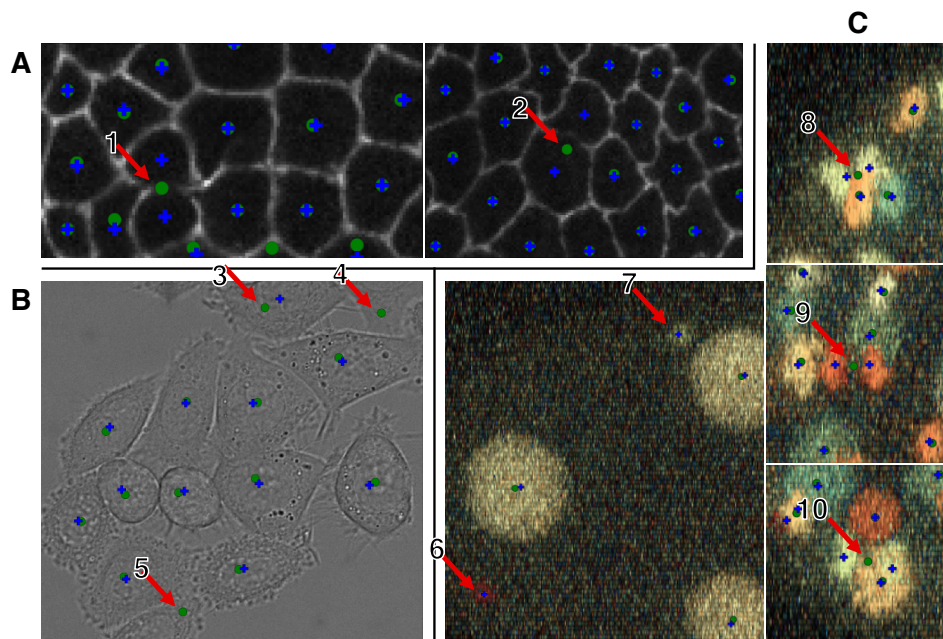


FIGURE 2.10 CP-Net can occasionally correct noisy labels, shifting abnormally off-center annotations and correcting false merges in the annotations across multiple datasets. **(A)** *D. melanogaster* wing with e-cadherin GFP. Markers (1,2) show how it is possible for CP-Net to correct false merges in the annotations. **(B)** DIC-C2DH-HeLa from the ISBI Cell Tracking Challenge. Marker (3) shows a particularly off-center annotation being re-centered. Markers (4,5) show annotations which don't appear to correspond to any cell creating (False) False Negatives. **(C)** Fluo-N3DH-CE *C. elegans* early embryogenesis with GFP-Histone rendered with false-colored, depth-encoding max projection to help distinguish overlapping nuclei. Markers (6,7) show small, apoptotic nuclei from the four cell stage which are un-annotated, but detected by CP-Net, and will be registered as (False) False Positives. Markers (8,9,10) show multiple occasions where cell divisions are annotated with a single point placed half-way between telophase nuclei while division is still occurring. CP-Net identifies telophase nuclei individually, often resulting in two False Positives and a False Negative during waves of cell division. (See section on *C. elegans*).

Most bioimage data including datasets from the ISBI Cell Tracking Challenge contain missing and erroneous annotations, i.e. *label noise*. Given

that label noise of some kind will be present in all manually annotated datasets we need models that are robust to such noise. Label noise present in the training set sends conflicting signals to the model, while label noise in the test set gives misleading numerical results which place an upper bound on the performance of a perfectly trained model. In cases of ambiguous data some amount of label noise is unavoidable. Challenges like the ISBI CTC take this into account by having multiple parties provide independent annotations of each dataset, which allows scoring annotations for consistency.

fig. 2.10 shows the predictions of a trained CP-Net model on label noise that was present *in the training set*. This demonstrates both the ability to correct label noise, as well as avoid overfitting on a variety of signals and data types.

2.4.3 SYMMETRIC NEAREST NEIGHBORS MATCHING SCORE

In order to evaluate the quality of centerpoint predictions we define the Symmetric nearest neighbors matching (SNNM) between point clouds and its associated f_1 , precision and recall scores. The matching is constructed by placing a directed edge from each point in the predicted set, P , to it's nearest neighbour in the ground truth set, GT , and vice versa. This creates a directed graph as shown in fig. 2.11 from which we define a matching between point pairs who mutually point to one another. This matching is fast to compute, requiring only $O(n \log n)$ operations when using a kd-tree for the nearest neighbour query, and does not require specifying an arbitrary cutoff distance, but scales naturally with point density. True positives correspond to matched point pairs, while false positives correspond unmatched objects from the proposal set (blue) and false negatives to unmatched objects from the ground truth set (green). This allows us to define an f_1 score on the SNNM, $|TP|/(|T| + |P|)$, as well as precision, $|TP|/|P|$, and recall, $|TP|/|T|$. We report the f_1 score whenever we have fully annotated ground truth point clouds and the recall whenever the data is only sparsely annotated.

2.4.4 EFFECT OF KERNEL SHAPE

We begin with a simple experiment to help us understand the role of kernel size hyperparameters. Our intuition might suggest that the optimal kernel size should somehow be related to the size of the objects we want to detect, but should it be smaller or larger? And how important is this relationship?

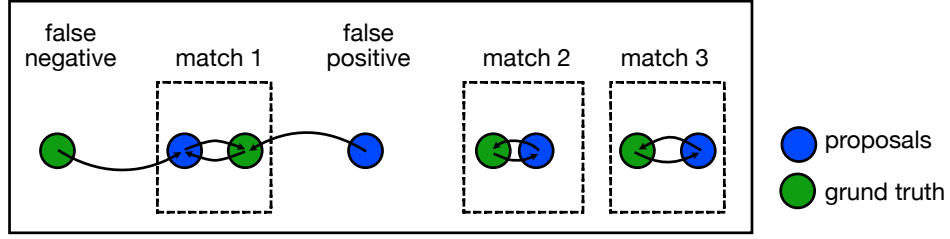


FIGURE 2.11 A definition of the Symmetric Nearest Neighbour Matching and associated F1 score. Objects are assigned to single nearest neighbour in the alternate set. Only symmetric neighbour relations, i.e. cycles in the graph, count as matches in this scheme. One detected object (blue) does not match symmetrically, making it a False Positive. One ground truth object (green) does not match symmetrically, making it a False Negative. Thus the SNNM-F1 score for this proposal is $\frac{2|TP|}{|T|+|P|} = 6/8$.

In fig. 2.12 we see that detection with CP-Net is surprisingly robust to changes in kernel size. We also observe that performance is more variable and convergence is more difficult when the kernel size is small, in particular for the GOWT1 dataset with relatively larger objects. For intermediate sized kernels we see that the network’s centerpoint predictions consistently achieve maximal performance, and the predicted kernel shapes more closely resemble the ground truth target. When target kernels are very large we see the network struggles to recapitulate the exact shape of the target kernel. Thus, even though the overlap between target and predicted kernel is substantial, the normally smooth peak of the kernel becomes rough with multiple local maxima, causing overdetection. While it may in theory be possible to correct this situation by using an alternate peak finding method (e.g. by fitting a parameterized smooth shape to the predicted kernel peaks) we prefer the speed and simplicity of local max filters, and simply avoid this problem by using appropriately sized kernels.

For both datasets the range over which we have nearly optimal performance is very broad, covering at least one order of magnitude between $[1,10]\text{px}$. This stability is convenient as it often allows us to select the optimal kernel size by eye on the first try. Beyond 10px performance decreases for both methods as noisy kernel shapes lead to overdetection.

The five panel columns correspond to models trained with kernel sizes $\sigma \in [0.44, 1.36, 4.23, 13.12, 40.69]$. For the smallest sizes, notice how the predicted kernel broadens significantly relative to the target, and even mirrors the shape of the cells in the raw input. For very large kernels the network struggle with overdetection, see ground truth annotations (green circles) vs CP-Net predictions (blue crosses).

Intuitively we might try to understand these results in the following way: Making the kernel smaller will make it *harder for the network to guess it’s*

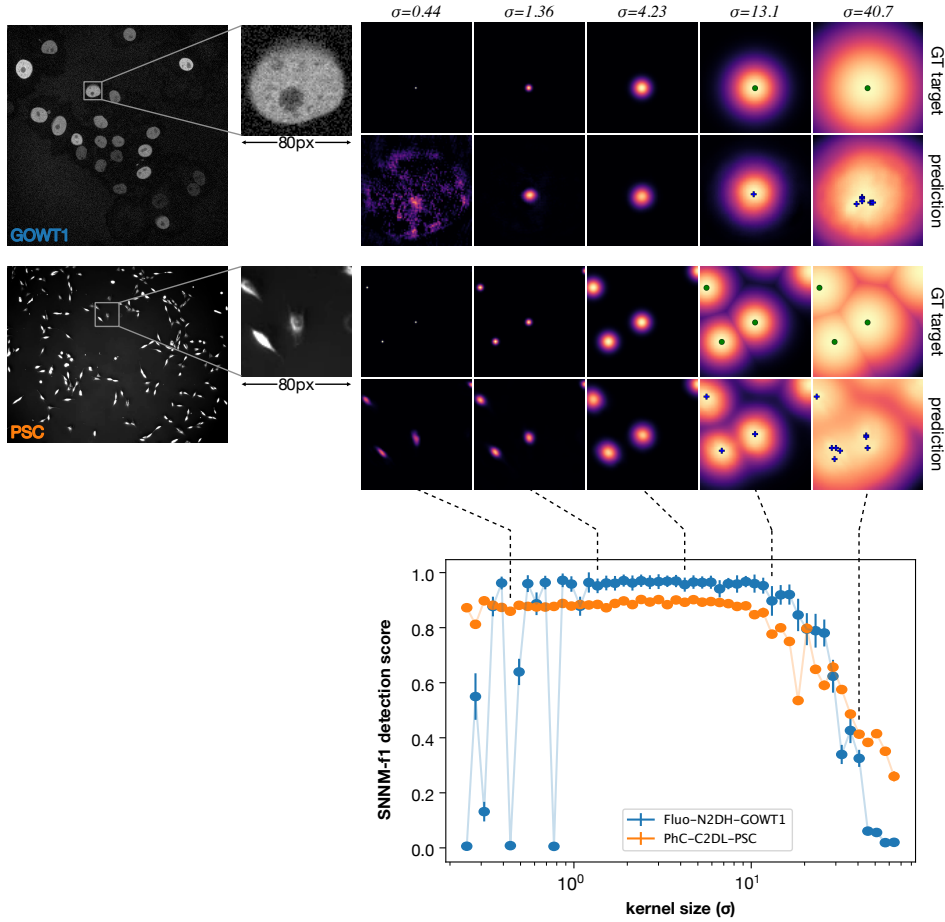


FIGURE 2.12 Effect of target kernel geometry and object geometry on the accuracy and stability of training. We compare detection quality as a function of training kernel size, σ , for the GOWT1 and PSC datasets. The relatively large cells in GOWT1 (blue) cause unstable training when the kernel size is small, but not for smaller cells in textttPSC (orange). The method is robust to choice of σ , with optimal performance over more than one order of magnitude. Green circle marks show ground truth centerpoints. Blue crosses are CP-Net predictions.

location (and thus harder for training to converge), but small errors in the predicted kernel shape are unlikely to lead to misdetections, as the very sharp peak of the kernel is hard to misinterpret by local max filtering. For large kernels we have the opposite effect, whereby small deviations in the predicted kernel location do not have a large effect on the loss, leading to more stable training, but *the network has a harder time predicting an accurate shape* which leads to multiple local maxima per kernel and thus overdetection.

2.4.5 EFFECT OF NOISE IN CENTERPOINT LOCATION

But what if our annotations are hastily made, imprecise or random? In fig. 2.13 we simulate imprecise annotations by adding random noise to

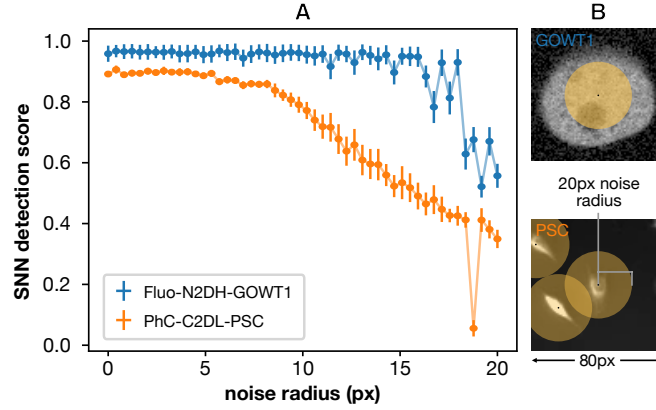


FIGURE 2.13 Label noise is simulated by adding a random displacement vector sampled from a circular disk (**B**) to each original centerpoint annotation before training. In (**A**) we plot the SSN detection score (evaluated on GT centerpoints from test set without noise) as a function of the radius of this disk for the GOWT1 and PSC datasets. While both datasets are quite robust against this additional label noise we observe that the larger cells of GOWT1 provide a stronger robustness against this noise in absolute pixel distances, although the robustness of PSC is particularly strong relative to its smaller average object size. Dot and vertical bar show mean and standard deviation evaluated across the 46 (GOWT1) / 67 (PSC) images comprising the test set.

the locations of the existing ground truth centerpoints. We observe that larger cells allow for sloppier annotations in terms of absolute pixel distances, but not relative to cell size. Note that the label noise added in this experiment is in addition to the (considerable) label noise that is already present in the data. This gives us confidence that we could not improve our performance by reannotating the dataset. The training for both datasets is performed with a fixed kernel radius of $\sigma = 4.23$, using 2D CP-Net trained with MSE loss.

2.5 DETECTING NUCLEI IN 3D

In this section we focus on detecting nuclei in images of developing organisms. Such data typically consists of many, densely packed cells in 3D in images with poor signal to noise ratio. Progress in microscopy techniques has spurred a renewed interest in developmental biology, in particular the introduction of *selective plane illumination microscopy* (SPIM) and *lightsheet* microscopy. However, imaging full organisms requires light to pass through biological tissue on both the illumination and emission paths, which results in poor image quality relative to single cell or *in vitro* systems. Yet these 3D images and timeseries of embryogenesis contain a wealth of information fundamental to developmental biology including how tissues and organisms obtain their shape as well as the mechanisms of self-organization and patterning in terms of single cells Keller et al., 2008; Keller et al., 2010; Faure et al., 2016; McDole et al.,

2018). Furthermore, the large scale (a single SPIM timeseries of a developing *Tribolium castaneum* can be several terabytes consisting of hundreds of timepoints each with thousands of cells) makes pure manual analysis too slow. These qualities have motivated the creation of a host of specialized approaches for detection, segmentation and tracking in embryogenesis data. However, despite their importance, bioimage methods for development lag behind methods for smaller, 2D datasets.

2.5.1 EXISTING DETECTION METHODS FOR EMBRYOGENESIS

Yasuda et al., 1999 was the first method to focus on automated analysis of timeseries images of developing organisms. They used a combination of classical filtering techniques including Kirsch and Prewitt operators, local histogram equalization, and local intensity variance characterization to detect nuclei in 3D+T differential interference contrast (DIC) images. Hamahashi et al., 2005 used local image entropy filters to improve detection of nuclei in these images, extending the tracking to the 24 cell stage.

In 2006 Bao et al., 2006 presented the first method which would perform the automatic detection and tracking of individual cells during a significant portion of *C. elegans* embryogenesis, with the goal of automated cell lineaging (detection and tracking of all cells during embryogenesis). Their program, *Starry Nite*, used 3D scanning confocal microscopy data with a fluorescent histone marker H2B-GFP to make the chromatin of individual cell nuclei visible against an otherwise dark background. *StarryNite* fits spheres to these bright 3D blobs to detect individual nuclei, processing each 3D stack individually. They report 99% detection accuracy through the 194 cell stage, but struggled at later stages with missed nuclei detections (false negatives) due to low signal intensity, high background noise, and cell crowding.

The emergence of lightsheet and SPIM microscopy in 2006 enabled biologists to study embryogenesis of a broader selection of model organisms at spatiotemporal resolution that made cell tracking feasible within certain windows of space and time. This led to attempts at cell lineaging in a variety of model organisms including zebrafish, *D. melanogaster* and early *M. musculus* embryos Keller et al., 2008; Keller et al., 2010; Khairy and Keller, 2011; Keller, 2013.

In 2014 Amat et al., 2014 introduced Tracking with Gaussian Mixture Models (TGMM), the first approach effective at tracking nuclei across multiple embryogenesis datasets from different organisms. TGMM uses a modified expectation maximization (EM) algorithm to fit a mixture of gaussians to H2B labeled nuclei in developing *D. melanogaster*, zebrafish,

and *M. musculus* which appear as densely packed bright blobs, often thousands of nuclei per image. These datasets each start at the 1k cell stage when nuclei are already somewhat homogeneous in size and appearance. TGMM is not designed to cope with large variability in nucleus size like is present in *C. elegans* dataset which starts at the one cell stage.

Bao et al., 2006 and Amat et al., 2014 were designed for fluorescence microscopy with bright, round nuclei on a dark background. We will show that CP-Net excels in this scenario, but also works with phase contrast, brightfield techniques, and with membrane signal.

They were not designed to work with other labels targeting e.g. the cell membrane or nuclear envelope. Nor were they designed to detect a wide variety of object shapes (e.g. fibroblasts, motile, or blebbing cells).

In summary, BioImage Analysis for embryogenesis employs heuristic algorithms which focus on specific cases, but thus far no approach has emerged that demonstrates good performance across a broad range of datasets. Here we show CP-Net takes step in this direction and how to deal with a variety of problems characteristic of developmental data.

2.5.2 C. ELEGANS NUCLEUS DETECTION

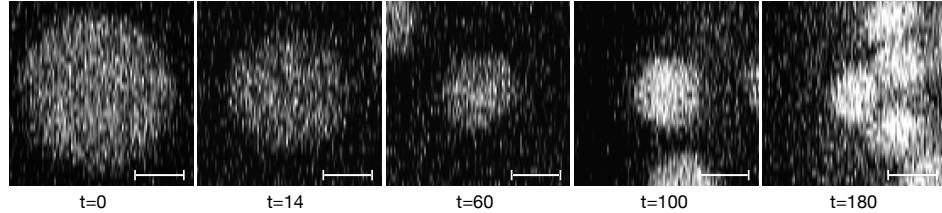


FIGURE 2.14 Average cell and nucleus size decreases over time as the embryo develops from a zygote to a hatched larva with over 500 cells. Here we show crops of single nuclei from the 4-cell stage until the 350-cell stage. White bar is 30px (3 μ m) across.

As cells divide in the early embryo of *C. elegans* they also decrease in size, as the total volume contained by all cells is fixed (see fig. 2.14). This causes a problem for CP-Net, as the target kernel size should be appropriate for the objects we want to detect, but manually adjusting the kernel size for each cell is tedious and defeats the purpose of using centerpoint dot annotations. In this section we address two important questions. Can CP-Net detect differently sized nuclei across the full timeseries of *C. elegans* embryogenesis, and how much labeled training data is required?

We attempt to answer these questions in fig. 2.15 where we compare four CP-Net models which differ only in their training data. When working with a new dataset in the wild we would like to do as little manual

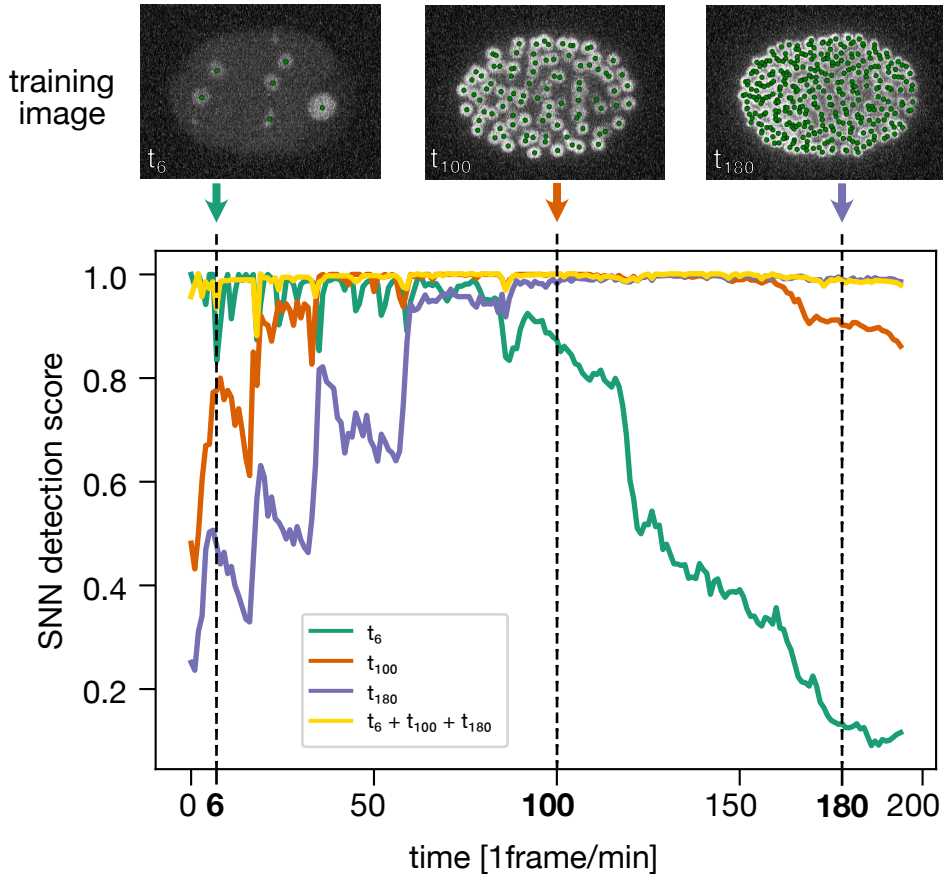


FIGURE 2.15 Plot shows SNNM-f1 detection score for three variations of CP-Net with distinct training data. **(left)** In each case we train with annotations from a single frame shown above. Despite the small number of labels CP-Net is able to generalize over significant periods of time during embryogenesis, yet no case is able to perform well over the entire time series. Both rapidly changing cell size/appearance and low absolute cell counts contribute to high early time variability. Note early time spikes correspond to waves of cell division.

annotation as possible, while simultaneously optimizing predictive accuracy. Thus, each of the first three models (t_{006} , t_{100} , t_{180}) uses for training only a single image selected from three qualitatively distinct embryonic stages, while the fourth (combo) uses the minimal amount of data necessary¹ to achieve excellent performance. We evaluate the performance of these models across every frame of the timeseries and notice that scores are highly locally correlated in time. As expected, none of the three single-frame models is capable of good detection performance across the entire timeseries. When we increase the amount of training data to seven diverse, hand-picked frames from across the full range (2,6,30,100,150,180,189) we see the model is capable of good performance across the board. Note the emphasis on a broad sample of objects leads to oversample the early timepoints and comes at the expense of a repre-

¹in our limited testing

sentative sample of all objects. However, perhaps most interesting is the good early-time generalization achieved by the t_{006} model despite only seeing a total of six centerpoint-annotated nuclei during training.

All models used the 3D CP-Net architecture and were trained in the standard way with MSE loss. Each trajectory shows an average over five models with identical parameters while varying initial kernel weights and random training patch sampling.

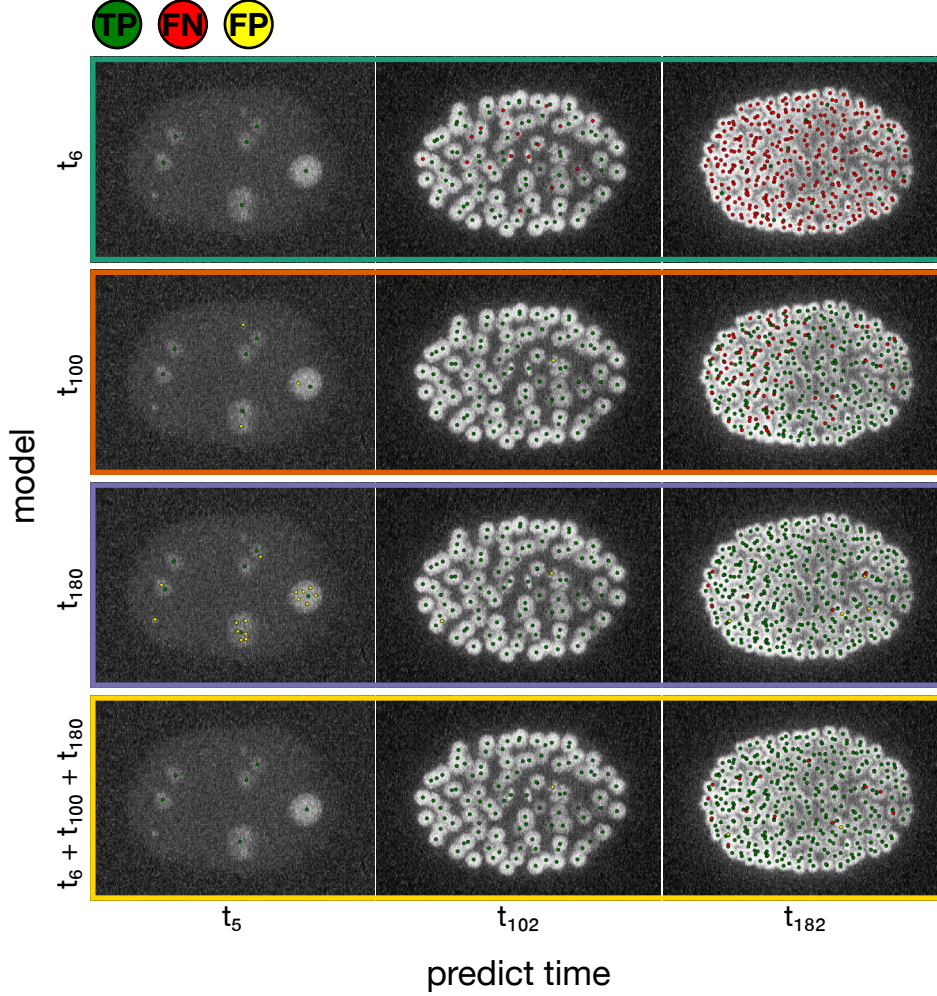


FIGURE 2.16 Models trained on a single timepoint fail to generalize to distant times. Training on small cells/late times causes us to over-detect on large cells/early times, while training on large cells causes us to miss the smaller cells at later times. We show the predictions from four different models (rows) which differ only in their training data, and their predictions on three different timepoints from a held-out test dataset (columns). Over-detections (false positives) are yellow dots, missed detections (false negatives) are red, and predictions with a match in the ground truth are green.

What is the cause of this poor generalization across time by the three single-frame models? We investigate this question in fig. 2.16 by looking at the point predictions by each of the four models from fig. 2.15 at three test times (not included in train or validation set). Poor performance

appears largely to be explained by the decrease in average cell size with time. Thus, models trained on later times and smaller cells *overdetect* when predicting on the large cells at the beginning of the timeseries. The reverse is also true; models trained on early times and larger cells *underdetect* when predicting on small cells later in the timeseries.

These experiments show CP-Net is capable of small generalizations across time and nucleus size, but that the window of generalization is only about a third of the total timeseries. We also show that the model is capable of learning from the union of the three training sets which appears to strictly improve performance across all times.

2.6 ISBI CELL SEGMENTATION BENCHMARK PERFORMANCE

We evaluate the CP-Net detection model by participation in the ISBI Cell Segmentation Benchmark which let us compare detection performance on a broad spectrum of bioimage modalities versus current state of the art approaches. The challenge hosts twenty datasets including bright field (BF), phase contrast (PhC), differential interference contrast (DIC), fluorescence (Fluo) microscopy images of cells and nuclei and including a wide selection of model systems. We entered submissions for as many datasets as possible. The challenge evaluates submissions on both segmentation and detection quality using SEG and DET scores. Over 50 groups have participated in the challenge since its inception in 2014.

Detection models based on learned convolutional features can be adapted to new imaging modalities with different low-level image statistics just by retraining. This gives CNNs an advantage in challenges with a broad selection of datasets.

CP-Net detects centerpoints only, but does not segment objects. So in order to submit to the challenge we extended CP-Net with the minimum viable segmentation routine, placing solid spheres at detection sites. The ISBI CTC requires that detection, segmentation and tracking results are saved as labeled instance segmentation images. To adhere to this format we first use CP-Net to detect objects across all images producing a set of detected centerpoints. Then we create a series of label images using these centerpoints where each object has the same size and shape, a sphere which has been rasterized to an anisotropic grid, and in cases of overlapping objects we assign a label of the nearest centerpoint to the pixel.

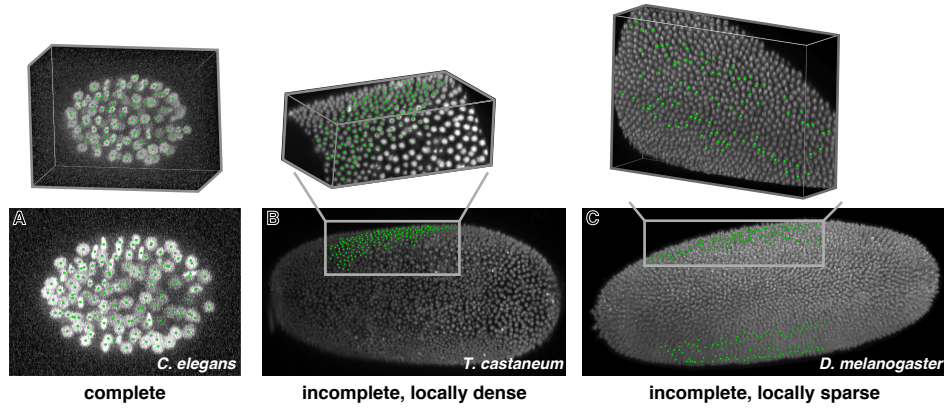


FIGURE 2.17 Point annotations from the ISBI datasets come in three styles determined by their completeness and density. (A) In completely annotated images every object has an annotation. (B) In incomplete but densely annotated images there are distinct regions of complete annotations. (C) Incomplete and sparsely annotated images have a small number of objects interspersed throughout image. The style of annotations determines how we generate the training data. It is still possible to learn from incomplete, sparse annotations if we use a loss which weights pixel proportional to the value of $\text{target}(x)$.

2.6.1 SPARSE VS DENSE ANNOTATIONS

One of the major problems in adapting learned detection approaches to 3D datasets is the difficulty of creating high quality annotations. In fig. 2.17 we categorize the annotated developmental datasets from the ISBIctc into three annotation styles: complete & dense, incomplete & dense, and incomplete & sparse.

This method of categorization is independent of the geometry or type of an individual annotation. In general the amount of information in an annotation (and the difficulty of creating it) ranges from complete & dense instance annotations down to *sparse & incomplete centerpoint annotations* fig. 2.2.

Dense instance annotations are very difficult to create in 3D for anything but small image crops. While dense annotations provide strictly more information per image, they may not be the most efficient use of annotator time! And often it is better to have a larger number of images with sparse&incomplete annotations than a small number of dense&complete annotated images. Centerpoint annotations have the advantage of being easier to create, but even still most datasets fall short of annotating every cell, leaving some areas within the image blank. When generating patches from an image with incomplete annotations we must take care that the un-annotated objects don't teach our network bad habits (mislead it into learning the wrong features). Many datasets which claim to be complete are actually unintentionally missing a small percentage of annotations, in particular in low SNR, 3D images where the annotation

process is extremely tedious. On the other hand, our annotator may simply choose to annotate only a fraction of the objects in the image for reasons of time and effort. Whether intentional or not, we can think of missing annotations as a kind of *label noise*, i.e. during training our network may see two very similar cells in the raw data that have conflicting annotations. Ideally, we would also be able to train our network in a way such that it acts as a kind of denoiser for our data, potentially even correcting sloppy or missing annotations within the dataset, (see fig. 2.10).

The simplest approach is to ignore these misannotations, and trust that our network will learn to correct them. This works when the number of missing annotations seen by the network is small and their labels are uncorrelated with learnable image features. Training with MSE teaches our model to converge towards the average $\text{target}(x)$ conditional on input over all training data. When the target values form a bimodal distribution around $\text{target}(x) = 1$ (correctly annotated) and $\text{target}(x) = 0$ (missing) our model predictions approximate the weighted average of the samples. So even with a large fraction of missing annotations we can still expect our model to produce predictions similar to target, but rescaled depending on the fraction of missing annotations.

To minimize label noise we ensure that each dataset’s annotations are sufficiently dense, i.e. that almost all of the objects seen in each patch have been labeled. And when the data is sparse, e.g. in Fluo-N3DL-DR0, we apply a pixelwise masking strategy to the loss to help defend against label noise.

It works by masking all pixels in the loss where target has a value less than some threshold. The ISBI Cell Tracking Challenge data contains three sparsely annotated datasets: Fluo-N3DL-DR0, Fluo-N3DL-TRIC, Fluo-N3DL-TRIF. We set the threshold to 3σ based on visual analysis of the density of cells,

2.6.2 TRAINING `cp-net` FOR ISBI CSB

We train only on centerpoint annotations and ignore the sparse instance segmentation gt. The data from both 01 and 02 acquisitions were pooled together, but otherwise the data for each of the 19 datasets were kept separate.

Our approach aims to minimize the number of parameters which must be changed across datasets in the hope of achieving a more robust and reusable detection methodology. We manually vary a single parameter (the average cell size) across each dataset, and use it to determine all other internal parameters automatically. We forbid the use of training

data from outside the ISBI CTC itself, and use no pretrained weights from existing architectures. All parameters are either trained from scratch, or determined automatically from a single, intuitive user-defined variable. We perform no hyperparameter tuning or optimization outside of the gradient descent used for training CP-Net .

The training procedure is adapted slightly from section 2.3 to deal with sparsely annotated datasets. For compact annotations we only keep training patches which have a significant fraction of nonzero target label. This treats the unannotated border regions like label noise. For truly sparse annotations we apply a mask in the loss at 3 sigma in every direction.

See table ?? for specific parameter values for each dataset.

2.6.3 VISUAL RESULTS ON ISBI CSB

In fig. 2.18, fig. 2.19, fig. 2.20 and fig. 2.21 we show the results of our model on 2D and 3D data from the ISBI Cell Segmentation Benchmark. We train a model for each dataset using only the centerpoint annotations provided in ISBI CSB training acquisitions 01+02 and apply the same fundamental training scheme described in section 2.3 for all datasets. As described, when data is sparsely annotated we use a pixelwise weighting scheme that emphasizes the loss only near annotated objects.

The colored circles mark the boundaries of segmented instances. The same size circle (oblate spheroid in 3D) is used for every object and placed at every centerpoint detected by CP-Net . We use the middle time-point from challenge acquisition 02 for all images, so none of the data shown in these figures have been seen by the model before. Pixel size is constant across panels in fig. 2.18 and fig. 2.19 and the figures maintain single pixel precision in PDF viewers which allow nearest-neighbour interpolation (e.g. Skim on MacOS). Min, max and gamma are manually adjusted for each panel to make cells and nuclei visible.

The results show that CP-Net functions across a broad spectrum of data types (all datasets available in the challenge). It is the only model participating in the challenge which has submitted results for every dataset. While the naive segmentation markers are a poor fit for many objects, they allow us to participate in the challenge using only a cell centerpoint detection model requiring zero cell segmentation ground truth for training.

For 2D segmentation results see fig. 2.18. For the 3D segmentation results we created panels that allow a clear view of segmentation quality from max projections through the full image volume. For datasets Fluo-N3DL-TRIF and Fluo-N3DL-DRO which are insufficiently sparse when

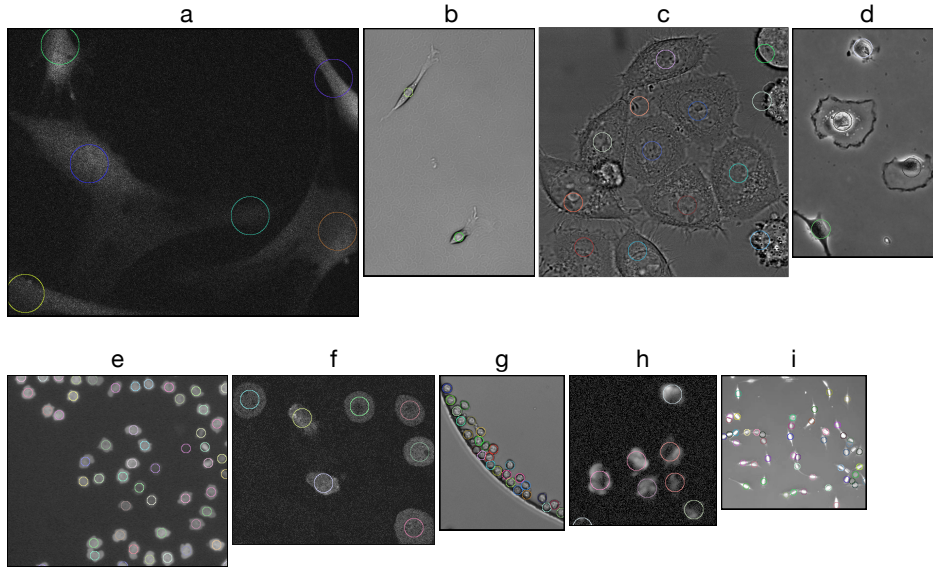


FIGURE 2.18 2D Segmentation results from the ISBI CSB, acquisition 02.

(a) Fluo-C2DL-MSc $t=24/48$. CP-Net struggles on these large, mesenchymal stem cells that lack clear borders. Small number of cell and ambiguous cases lead to poor scores. (b) BF-C2DL-MuSC $t=688/1376$. CP-Net detects non-round cells well here and avoids false positives despite petri dish background texture with cell-like curves. (c) DIC-C2DH-HeLa $t=57/114$. The training annotations are not centered, especially for cells near boundaries. This causes off-center predictions which may go unpenalized. (d) PhC-C2DH-U373 $t=57/114$. Small number of total annotations. Cells have wildly different morphologies, but CP-Net finds the center. (e) Fluo-N2DL-HeLa $t=46/92$. Hard to distinguish two touching cells from large single cell. The low contrast between dim cells and background causes some FP and FN. Panel uses $\gamma = 0.36$ to help show dim cells. (f) Fluo-N2DH-GOWT1 $t=46/92$. Small number of cells, well separated. CP-Net is very good at dealing with low contrast and noisy background. We use $\gamma = 0.5$ to display dim cells. (g) BF-C2DL-HSC $t=882/1764$. Very small, uniform cells with clear, dark boundary signal. CP-Net has no problem with densely packed cells and avoids false positives on petri dish edges or background trash. (h) Fluo-N2DH-SIM+ $t=69/138$. Simulated data with weak object boundaries. It would be hard to come come up with unambiguous instance segmentation labels, but the centerpoint is still intuitive. Sometimes individuals look like multiple smaller objects combined. Gamma of 0.5 to show dim cells. (i) PhC-C2DL-PSC $t=150/300$. Very small, elongated, objects. Cells are oversaturated making it hard to find contrast between densely packed neighbors. No way to find boundaries between touching objects except for via prior-knowledge of cell shape.

projected along the Z dimension we created separate figures fig. 2.20 and fig. 2.21 where we randomly sampled individual nuclei to allow for better visual inspection.

2.6.4 SCORES ON ISBI CSB

In table 2.1 we show CP-Net detection results (DET scores) across all datasets and compare to three alternative methods. The methods in comparison are the top three which maximize the sum total of DET scores

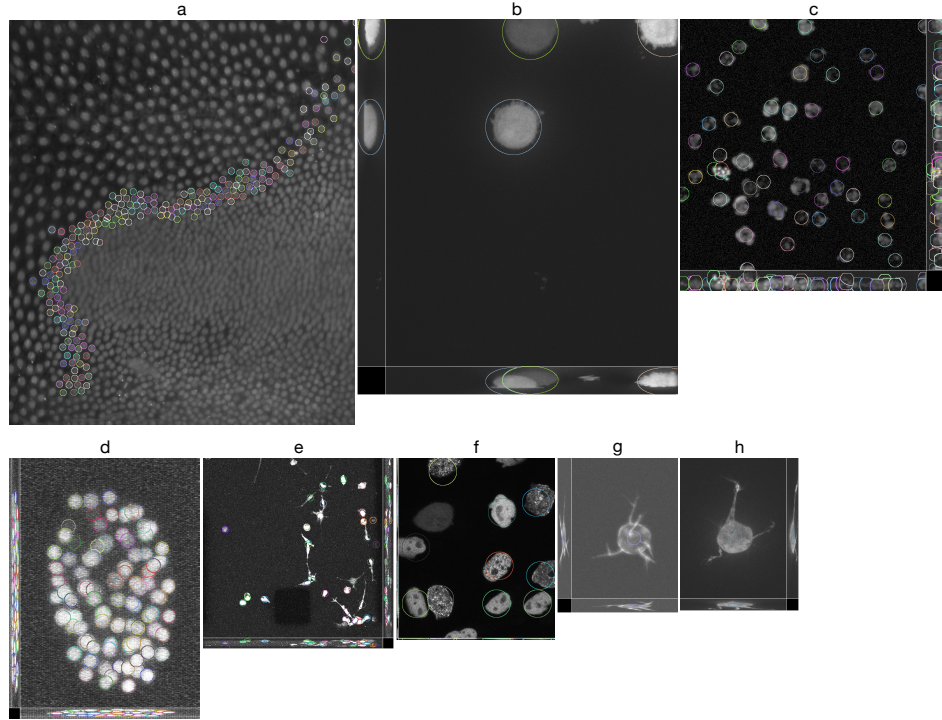


FIGURE 2.19 3D Segmentation results from the ISBI CSB, acquisition 02. Each panel features XY, YZ and XZ views with original anisotropic Z resolution.

(a) Fluo-N3DL-TRIC $t=85/170$. Very thin, quasi 2D data. CP-Net achieves state of the art detection on these uniform, round cells despite low contrast and high packing density. Note according to ISBI convention only the subset of detected cells which have GT annotation in the first timepoint are highlighted. (b) Fluo-C3DH-H157 $t=30/60$. Sparse, round cells with clear boundaries. (c) Fluo-N3DH-SIM+ $t=55/110$. Simulated cells which lack clear boundaries and have very low signal to noise ratio. Panel shows overdetected on apparently apoptotic cell with many bright foci. (d) Fluo-N3DH-CE $t=70/140$. Data is highly anisotropic with 11/1/1 voxel aspect ratio. The nuclei are uniform and round so the oblate spheroid model is a good fit for middle times, but a poor fit at early and late times as nuclei multiply and shrink. Image quality decreases with Z depth, but CP-Net detects all nuclei even with very bad SNR at deep Z with state of the art precision. (e) Fluo-C3DL-MDA231 $t=6/12$. Very difficult dataset for CP-Net. Low annotation count, non-uniform and elongated, branching shapes. Cells are heavily oversaturated and visual boundary cues on cell clusters are hard to find / require looking at object geometry. Background is noisy and high relative to the most dim cells. (f) Fluo-N3DH-CHO $t=46/92$. Large, flat cells with interesting textures. Quasi 2D data again. (g) Fluo-C3DH-A549-SIM $t=15/30$. (h) Fluo-C3DH-A549 $t=15/30$. GFP-actin-stained A549 Lung Cancer cells. CP-Net avoids false positive detections on cell protrusions.

across all datasets, i.e. methods which are broadly applicable. Across all 17 nontrivial datasets CP-Net has a sum total DET score of 14.8. UP-PT totals 9.893 using Laplacian of Gaussian (LoG) filter for nucleus detection but struggles with noisy, low contrast imagery. KIT-Sch-GE totals 11.341 using a 2D slice based instance segmentation with CNNs, but rely on large amounts of full instance segmentation training data from outside of the challenge. KTH-SE-1 totals 9.506 using classical denoising and smoothing heuristics plus thresholding segmentation, but focuses almost

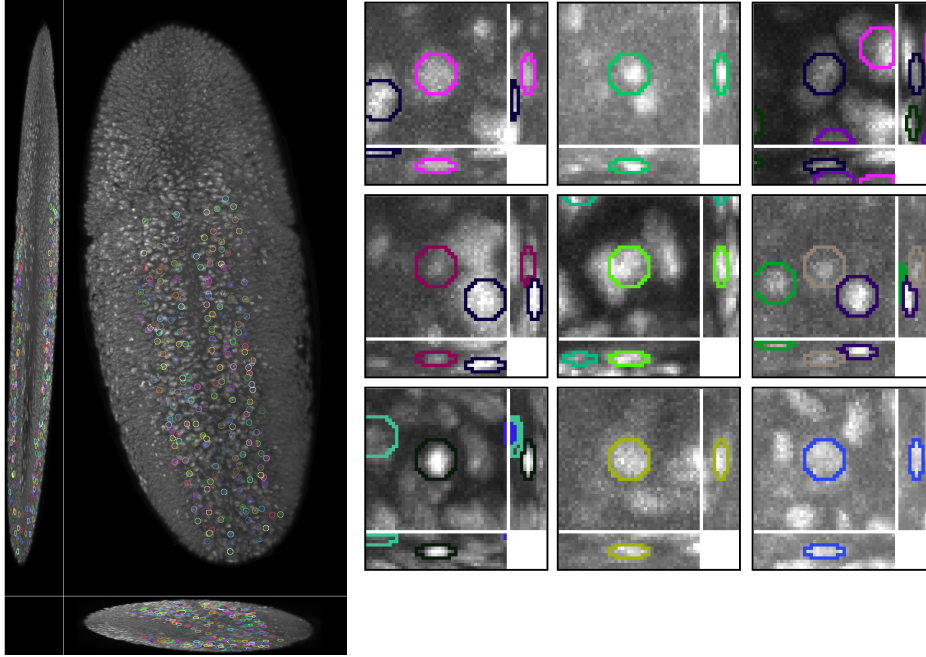
Drosophila melanogaster

FIGURE 2.20 3D Segmentation results from the ISBI CSB Fluo-N3DL-DRO acquisition 02 timepoint 25/50. Only segmentation label boundaries are shown. We randomly sampled nine individual predicted instance labels from the image for in depth inspection. The labels displayed in whole-embryo panel above are only the subset corresponding to the manual annotations given in the $t = 0$ ground truth.

exclusively on fluorescence microscopy data and requires specialized optimization of 17 parameters. Finally, the current best score on each dataset is shown in column "Rank #1" which achieves a 16.473 out of a total of 17.

Also, CP-Net is competitive with the current best performing method in a few categories, achieving a DET score within the top three methods on three datasets, Fluo-N3DH-CE (1st), Fluo-N3DL-DRO (3rd) and Fluo-N3DL-TRIC (1st).

The CNNs enable this broad spectrum performance. Other learning based methods typically use large amounts of external data for training. All methods are allowed to visually overfit to the challenge data. This is easier if you have hyperparameters to tune.

In figure fig. 2.22 we show the dependence of CP-Net DET scores on the total number of annotations available for each dataset. The dependence is very clear, and suggests a high upper bound on detection quality, and that the number of available annotations for a given dataset is the limiting factor for detection quality in most cases.

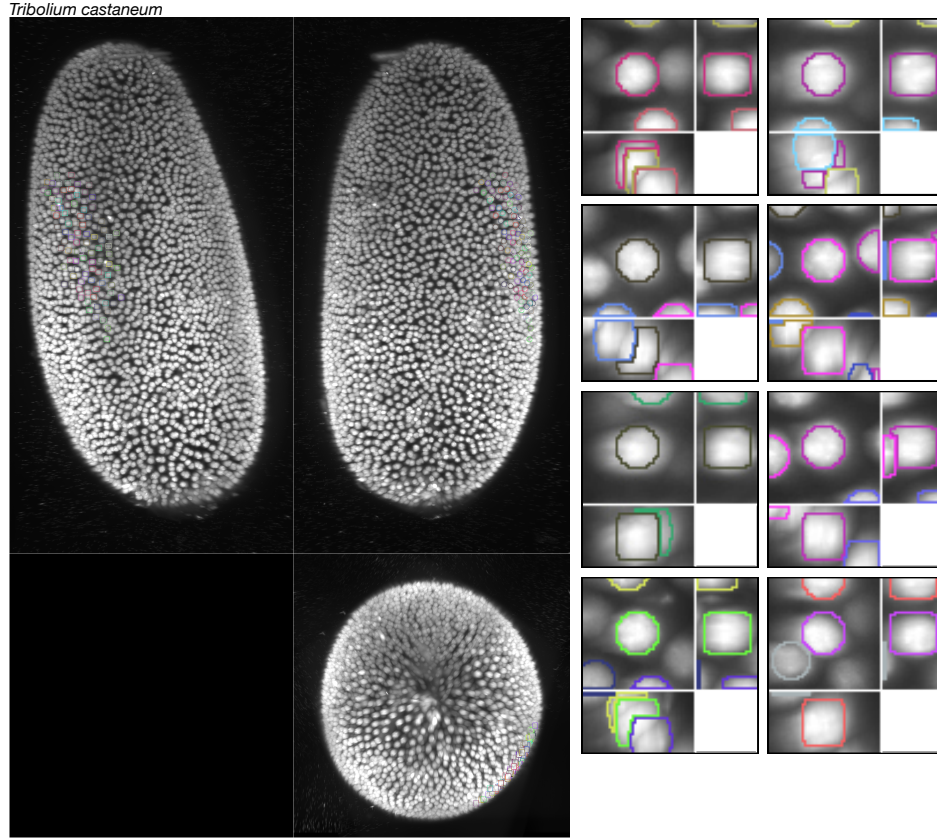


FIGURE 2.21 3D Segmentation results from the ISBI CSB, Fluo-N3DL-TRIF acquisition 02, timepoint 38/76. Only segmentation label boundaries are shown. We randomly sampled eight individual predicted instance labels from the image for in depth inspection. The labels displayed in whole-embryo panel above are only the subset corresponding to the manual annotations given in the $t = 0$ ground truth.

2.7 CONCLUSION

In this chapter we introduced CP-Net a CNN based method for cell and nucleus detection which requires only single-dot centerpoint annotation for training. We showed in experiments on datasets GOWT1 and PSC that CP-Net is robust with respect to annotation noise in centerpoint location (fig. 2.10) and changes in the kernel size hyperparameter (fig. 2.12). We showed on various datasets that CP-Net can correct small numbers of wrong, missing, or ambiguous labels.

We demonstrated its applicability on a number of 2D and 3D BioImage datasets across fluorescence (SPIM and confocal), phase contrast, differential interference contrast, and brightfield microscopy. Additionally, we showed how to adapt CP-Net to work in case of sparsely annotated datasets.

The high quality achievable by classical heuristics is impressive. It would be interesting in future work to compare the training procedures of the

Dataset	Rank #1	UP-PT	KIT-Sch-GE	KTH-SE-1	CP-Net
BF-C2DL-HSC	0.988	—	0.974	—	0.985
BF-C2DL-MuSC	0.982	—	0.977	—	0.927
DIC-C2DH-HeLa	0.961	0.577	—	—	0.675
Fluo-C2DL-MSC	0.876	0.482	—	0.851	0.254
Fluo-C3DH-H157	0.988	0.928	0.949	0.988	0.976
Fluo-C3DL-MDA231	0.907	0.788	0.851	0.907	0.767
Fluo-N2DH-GOWT1	0.980	0.862	0.950	0.976	0.907
Fluo-N2DH-SIM+	0.981	0.901	0.949	0.960	0.846
Fluo-N2DL-HeLa	0.992	0.974	0.992	0.992	0.984
Fluo-N3DH-CE	0.959	0.579	0.930	0.959	0.990
Fluo-N3DH-CHO	0.954	0.819	0.945	0.954	0.883
Fluo-N3DH-SIM+	0.969	0.806	0.937	0.953	0.912
Fluo-N3DL-DRO	0.988	0.375	—	—	0.962
Fluo-N3DL-TRIC	0.994	—	0.961	—	0.994
Fluo-N3DL-TRIF	0.997	—	0.926	—	0.986
PhC-C2DH-U373	0.985	0.889	—	—	0.817
PhC-C2DL-PSC	0.972	0.913	—	0.966	0.943

TABLE 2.1 Rank #1 shows state of the art DET scores on ISBI CSB as of March 2021. We compare CP-Net DET scores along with the top three performers as ranked by sum total of DET scores across all datasets. Method scores equal to or greater than Rank #1 are in bold.

CNN based and classical heuristic based methods in terms of time, stability, and data efficiency.

The class of CNN models is itself quite broad, and varies in the task performed and training data required. By using centerpoint annotations CP-Net has access to a larger pool of annotations without introducing new data from outside the challenge, either via simulations or new hand annotation. But it remains an open question whether these "weaker" annotations are truly more efficient than the larger, more difficult full instance label annotations in terms of the total annotation time required to achieve a given detection quality. In future work it would be interesting to compare this annotation time directly while keeping the task and dataset constant.

The total number of annotations is a critical variable in our method, limiting quality in many cases. However, we have not explored much by way of data augmentation techniques. It would be interesting to see if we could enhance the quality of CP-Net predictions by introducing, e.g. continuous rotations, noise augmentation, cutout, etc.

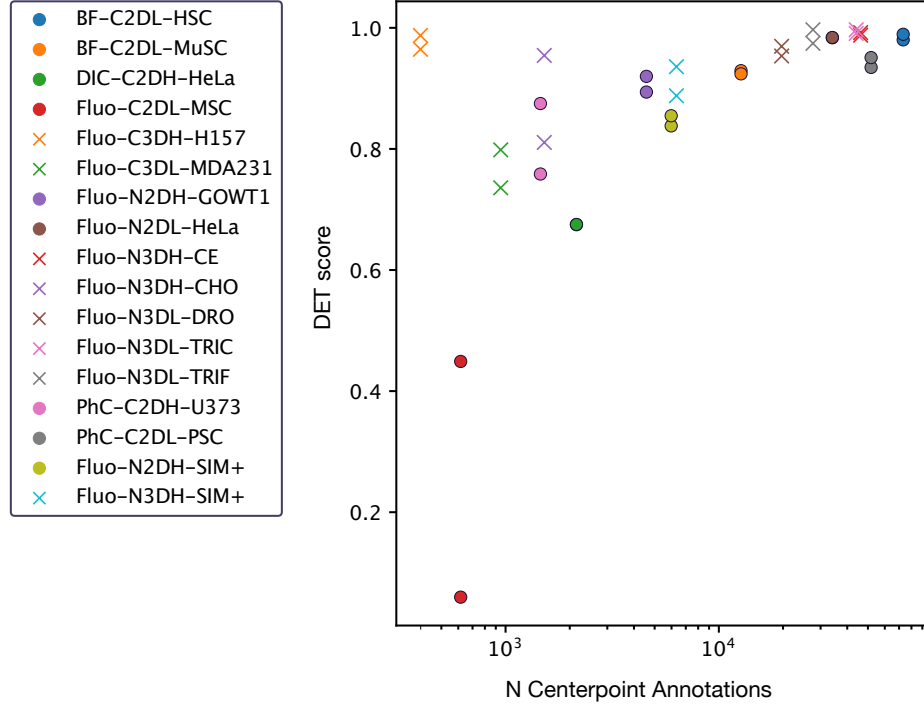


FIGURE 2.22 The number of annotations available for training CP-Net has a strong impact on detection performance. The correlation shown here suggests that the number of annotations is performance limiting for most cases, not model capacity. We perform standard mirroring and discrete $\pi/2$ XY rotations as augmentation for each dataset, which increases the effective size by a factor of 8 (2D) and 16 (3D) due to the additional Z-mirroring. Each dataset has two associated acquisitions and we plot their scores separately.

Finally, we believe CP-Net demonstrates good performance across diverse datasets with minimal parameter tuning due to solid choices with respect to architecture, task and sampling procedure. These choices allow CP-Net to operate with reasonable efficiency across a large number of datasets without any changes to these fundamental hyperparameters.

Tracking cells and nuclei through time has become a critical task in biology as it provides insight into cell lineage and cell fate determination as well as the physics of cell motion, Eils and Athale, 2003; Meijering et al., 2009; Aigouy et al., 2010; Icha et al., 2016. However current automated methods of cell tracking are only accurate across a fraction of available datasets, even with highly tuned, data-specific solutions. In particular tracking models struggle on 3D+t movies of developing organisms where hundreds or thousands of cells move in an intricate, coordinated fashion as large imaging volumes and light-sensitive samples necessitate low frame rates and low spatial resolution.

In part one of this chapter we explore the limits in quality for the simplest possible tracking model: *nearest neighbour tracking*. Encouraged by our results on cell detection in the previous chapter, we combine the CP-Net detection model with simple nearest neighbour linking to track cells in various datasets from the ISBI Cell Tracking Challenge. Our results show competitive tracking performance across a broad spectrum of datasets despite the simplicity of the linkage model, including 1st place tracking scores¹ in each of the four datasets of developing embryos including significant reduction in SoA error rate in most cases. We provide additional experiments on these datasets to estimate the maximum tracking quality achievable with nearest neighbour tracking for the case of perfect detection. Together, these results demonstrate that it is possible in most cases to achieve high quality tracking by increasing the accuracy of the detector without complex modeling of cellular motion. This has implications for determining the right tradeoff between model sophistication, implementation complexity, and runtime of linking models.

In part two of this chapter we introduce a novel extension to the tracking by assignment paradigm (Kausler et al., 2012) for the purpose of including prior knowledge of spatial correlations in cell motion, e.g. tissue flow during gastrulation, muscular contractions, or cell crowding. We perform

¹tied for first place on the F1uo-N3DL-Dro fly dataset

experiments on two datasets exhibiting correlated motion of neighbouring cells and demonstrate significant improvements in accuracy compared to standard nearest neighbour and tracking-by-assignment baselines.

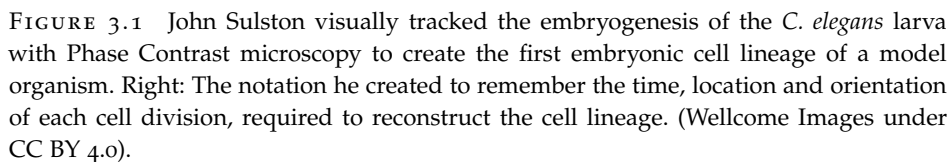
Contributions

1. Evaluate a simple nearest neighbour tracking model + CP-Net detection in the ISBI Cell Tracking Challenge (CTC), leading new SoA tracking scores on multiple datasets.
2. Introduce a new cell tracking model which incorporates a spatially coherent movement prior into the well studied *tracking by assignment* framework.

3.1 RELATED WORK

When, in 1983, John Sulston tracked the 3D motions of the hundreds of dividing cells during *C. elegans* embryogenesis (Sulston et al., 1983) he did not have the luxury of digital imagery allowing him to scroll back and forth through time, specialized software like Fiji, TrackMate and MaMuT (Schindelin et al., 2012; Tinevez et al., 2017; Wolff et al., 2018) for annotating cells in images, or automated tracking models like StarryNite or TGMM (Bao et al., 2006; Amat et al., 2014) to jump start the process. These technologies have made it possible to track large numbers of cells through their complex 3D motions, yet still cell tracking remains highly labor intensive and poses a significant bottleneck to our understanding of many important biological processes like embryogenesis, cell fate decision making and regeneration (Amat and Keller, 2013).

In 2006 Zhirong Bao in the lab of Robert Waterston released StarryNite (Bao et al., 2006), the first broadly successful attempt at fully automatic 3D tracking of cells during embryogenesis. Their approach to detection relied on the ellipsoidal appearance of cell nuclei in fluorescence images when viewed as a slice through the Z dimension. Tracking began with a nearest neighbour linking approach, followed by a heuristic correction process when three or more children are mapped to the same parent based on parent nucleus shape (mitotic nuclei are no longer spherical) and age (successive divisions separated by less than N frames are unlikely). The program, in a fully automated mode, was able to achieve good performance up through the 194-cell stage with reported detection and division accuracy of 99%/97% respectively, but struggled at later stages with missed nuclei detections (false negatives) due to low signal intensity, high background noise, and extreme cell crowding.



In 2012 and 2013 Bernhard Kausler and Martin Shiegg in the lab of Fred Hamprecht developed *tracking by assignment*, an extended graphical models for object tracking to the case of an unknown number of dividing targets, making the model suitable for tracking cells in microscope videos Kausler et al., 2012; Schiegg et al., 2013. This family of models is composed of an initial detection phase followed by the global optimization of an intricate graphical model capturing the relative probabilities of cell movement, division, and apoptosis as well as incorporating and correcting for potential undersegmentation resulting from the detection phase. Inference in this graphical model correspond to the solution of an *integer linear program* (ILP), an NP-Hard but well studied computational problem. The tracking by assignment paradigm was then extended in 2014 to include new data-specific constraints by Florian Jug such as positional order in the case of bacteria artificially constrained to move only in a single

spatial dimension Jug et al., 2014. Corinna Blasse and Florian Jug would continue to push tracking by assignment through the introduction *graph morality constraints* in 2016 for the case of membrane tagged epithelial cells which may be heavily over segmented, allowing the joint optimization of both cell segmentation from cell fragments and cell tracking to a global maximum across time, Jug et al., 2016.

In 2014 Fernando Amat in the lab of Eugene Myers released Tracking with Gaussian Mixture Models (TGMM) which would be the first widely cited tracking model performant enough to track thousands of cells in real time and capable of generalizing across multiple model species, (Amat et al., 2014). The model generates a compressed, supervoxel representation of the data then greedily fits unnormalized Gaussians to fluorescent nuclei. Nucleus position and orientation is extrapolated from the previous time point and relaxed to a new optimum using Expectation Maximization cite as well as heuristics to decide when the number of Gaussians (nuclei) in the model should increase or decrease.

In 2015 Klas Magnuson in the lab of Helen Blau published an approach to tracking cells in which optimal “tracklets” following the motion of an individual cell are constructed via the Viterbi algorithm and then added greedily to the full solution, (Magnusson et al., 2015). This approach, combined with heuristic cell and nucleus detection based in classical image filters was applied to all datasets in the ISBI Cell Tracking Challenge and remains the best performing method on average even six years later, despite the rise of machine learning and the increasing number of challenge participants.

In 2018 Jan Funke of Janelia Farms and Lisa Mais in the lab of Dagmar Kainmüller showed how deep learning techniques designed for the problem of 3D neuron segmentation in electron microscopy images (EM) can be used to segment and track tightly packed 2D epithelial cells with fluorescent membrane by treating it as a pure 3D segmentation problem resulting in cell tracks through spacetime that resemble a cascade of branching neurons, (Funke et al., 2018). Also in 2018 Christian Payer of TU Graz offered a similar approach to fully differentiable tracking of 2D cells via 3D segmentation, (Payer et al., 2018). They use recurrent networks to predict a high dimensional vector for each pixel which is encouraged to diverge from its neighbors along object boundaries through a *cosine embedding loss*. Object instances are then segmented jointly in space and time using a fast clustering algorithm Ester et al., 1996. Despite these successes cell tracking is still treated predominantly as a discrete problem utilizing more traditional tools from computer science and optimization.

3.2 NEAREST NEIGHBOUR TRACKING

Methods for cell tracking are often categorized into *detect-then-link* approaches such as Bao et al., 2006; Bise et al., 2011; Magnusson et al., 2015; Kausler et al., 2012; Schiegg et al., 2013 and *model-evolution* approaches such as K. Li et al., 2008; Smal et al., 2008; Amat et al., 2014; Dzyubachyk et al., 2010. Detect-then-link approaches enable the use of pre-existing object detection or segmentation models which can be applied independently to each time point followed by a linking step in which physical actions (e.g. movement, division, appearance, apoptosis) are assigned as relationships between detected cells. In every case the goal of cell tracking is to obtain a graph of relationships (edges) between cells (vertices) over all timepoints.

The nearest neighbour tracking method is a detect-then-link approach that works as follows:

For every detected object form a vertex in the graph. For every pair of consecutive times $(t, t + 1)$ and for every detected object d in frame $t + 1$ add a directed edge $d' \rightarrow d$ pointing to d from it's nearest Euclidean neighbour d' in the preceding frame t .

This algorithm is about as simple as one could imagine for the task of cell tracking, but it has a number of undesirable properties. It allows for a parent cell to have more than two daughters, and for spontaneous apoptosis even in very young cells, it doesn't allow for spontaneous cell appearance through image boundaries or for re-connecting a cell lineage when it is broken due to temporary misdetection². However, it has the advantage of being parameter free and fast when using appropriate data structures, e.g. KD-Trees, for the nearest-neighbour query. These properties, in addition to the simplicity of implementation, make nearest neighbour tracking a useful approach to linking in datasets with low spatial density or slow moving cells. In fact, if we could rely on the detector to locate objects perfectly, then nearest neighbour tracking would correctly link each pair of cells whose displacement between frames was less than half the distance to its nearest neighbour in the preceding frame. Thus, we ask the question, *what fraction of cell displacements in real datasets actually meet this condition?* This will give us an upper bound on the linking performance we can expect from this simplest-possible linking approach.

²Misdetections are even sometimes unavoidable, e.g. when using fluorescent lamin nuclear envelope markers during breakdown of the nucleus in metaphase the signal may disappear completely.

3.2.1 UPPER BOUND ON LINKING PERFORMANCE

We can answer this questions by using Ground Truth (GT) tracking data from the ISBI CTC, keeping detections fixed, but removing GT links. Then we re-link the GT detections using the nearest-neighbour linking algorithm. The re-linked solution is compared against the GT using the TRA tracking metric.

THE TRA TRACKING SCORE

In order to score the performance of of a tracking method we use the TRA score. The TRA score was originally employed by the ISBI Cell Tracking Challenge (CTC) in 2014 as a way to determine tracking quality that took into account the labor required to manually correct an imperfect solution. The Cell Tracking Challenge consists of 19 2D and 3D datasets with acquisitions spanning both cell and developmental biology using bright-field, differential interference contrast, phase contrast, and fluorescence microscopy. The set of tasks is the same across each datasets: detect, segment and track every cell. Ground truth detection, segmentation and tracking labels are provided for two “training” timeseries, with official scores handed out for performance on two test datasets which have no accompanying ground truth. This challenge has been ongoing since 2014, with scores updated every month and publicly available and visible at celltrackingchallenge.net.

The standard TRA tracking measure (Matula et al., 2015) is a weighted average of detection and linking errors derived from the *Acyclic Oriented Graphs Matching Measure*, (AOGM), (Matula et al., 2015),

$$\text{TRA} = 1 - \min(\text{AOGM}, \text{AOGM}_0) / \text{AOGM}_0 \quad (3.1)$$

with

$$\begin{aligned} \text{AOGM} &= 5.0\text{NS} + 10.0\text{FN} + 1.0\text{FP} + 1.0\text{ED} + 1.5\text{EA} + 1.0\text{EC} \\ \text{AOGM}_0 &= 10.0\text{FN} + 1.5\text{EA} \end{aligned}$$

using variable semantics,

- *NS* non-split (undersegmentation)
- *FN* false negative detections (node addition)

- *FP* false positive detections (node deletion)
- *ED* edge deletion (false positive edges)
- *EA* edge addition (false negative edges)
- *EC* edge change semantics (a division to a translation or vice versa).

The GT is represented by an image containing small rectangles placed at the center of each ground truth object and with a pixel value corresponding to a unique ID for each object. The relationships between objects are stored in an auxiliary piece of data that defines the lineage tree. A relation is defined between the set of proposed objects and the set of ground truth objects by connecting each pair where more than *half* of the GT marker pixels overlap with the proposal. When the relation between a proposed object and the ground truth is 1-to-1 then we have a *true positive*; when this relation 1-to-many then we have a *non-split* proposal. 0-to-1 and 1-to-0 mappings correspond to *false negatives* and *false positives* respectively. The resulting score is bounded, $\text{TRA} \in [0, 1]$, with a value 1 received for perfect agreement between the proposed tracking solution and the GT.

EXPERIMENTS ON GROUND TRUTH CENTERPOINTS

Using data from the Cell Tracking Challenge (CTC) we were able to compare the TRA scores obtained using nearest neighbour linking techniques using ground truth from the training set with those of the best publicly available methods on the highly similar, but not identical, test datasets. Three of the nineteen datasets (Fluo-N3DL-DR0, Fluo-N3DL-TRIC, Fluo-N3DL-TRIF) only provide sparse tracking annotations, thus decreasing the apparent density of cells and making the task of linking ground truth objects artificially simple. The datasets Fluo-C3DH-A549 and Fluo-C3DH-A549-SIM were removed as they only contain a single cell and therefore are guaranteed perfect TRA scores given ground truth cell locations no matter what linking method is used. In the table below table 3.1 we show TRA scores on the remaining 14 training datasets, along with current state of the art tracking scores from celltrackingchallenge.net which is an average over two withheld test datasets. We additionally include a baseline in which links are randomly assigned between pairs of cells from consecutive timepoints. Note, the TRA score is a *weighted* sum over errors. And in particular it's important to remember that the punishment for False Negatives is much greater than that for False positives. For this reason the lower bound on TRA resulting from random linking is actually quite high for all datasets when we use GT detections, as there are zero costly FN detections.

Dataset	Rank #1	Nearest Neib	Random Links
BF-C2DL-HSC	0.978	0.9958	0.8079
BF-C2DL-MuSC	0.971	0.9900	0.8311
DIC-C2DH-HeLa	0.955	0.9981	0.8020
Fluo-C2DL-MSC	0.873	0.9942	0.8238
Fluo-C3DH-H157	0.987	0.9907	0.8523
Fluo-C3DL-MDA231	0.882	0.9944	0.8048
Fluo-N2DH-GOWT1	0.979	0.9983	0.7937
Fluo-N2DH-SIM+	0.975	0.9994	0.7917
Fluo-N2DL-HeLa	0.991	0.9982	0.7855
Fluo-N3DH-CE	0.945	0.9987	0.7844
Fluo-N3DH-CHO	0.953	0.9944	0.8143
Fluo-N3DH-SIM+	0.967	0.9997	0.7923
PhC-C2DH-U373	0.982	0.9979	0.8181
PhC-C2DL-PSC	0.966	0.9993	0.7847

TABLE 3.1 Rank #1 corresponds to state of the art TRA score from the ISBI CTC. Nearest neighbour linking connects each ground truth centerpoint annotation at t to its nearest neighbour at $t - 1$. Random linking connects each centerpoint at t to a random centerpoint at $t - 1$. Scores are an average over 01 and 02 acquisition datasets.

Nearest neighbour (NN) tracking on ground truth (GT) centerpoints allows us to determine the relative importance of detection vs linking steps in two-phase tracking models. We find that nearest neighbour linking on GT centerpoints obtains near perfect TRA scores on every dataset. We include the top TRA score from the leader board at celltrackingchallenge.net, which shows real cell detection and tracking performance on hidden challenge data.

The baseline TRA score is very high (around 0.8) even for random linking on GT detections, because TRA penalizes heavily for missed detections.

There are a few lessons to learn from this experiment if we allow ourselves to assume that the training data is representative of the hidden challenge data. First, the relatively high scores of the nearest neighbour based techniques tell us that cells in these data are either not very densely packed, or they don't travel very far in between frames, thus intricate models of cellular motion are not required for high TRA scores. Complex motion models may still be useful if we e.g. wish to focus on a particular subset of rare events, e.g. on cell divisions where the motions of cells are more rapid. Further, we cannot conclude from these data that nearest neighbour methods are on their own *sufficient* for good TRA scores, but rather that **the focus of linking models in these data should be on correcting detection errors** rather than modeling and predicting cellular motion and appearance. Finally, we can not rule out that for more densely packed and fast-moving cells (e.g. developing embryos during gastrulation) models of cellular motion or appearance may indeed be necessary.

Additional implications are that we can improve TRA scores simply by improving our detector, almost indefinitely. This also has implications for the tradeoff between time spent in creation of detection and tracking ground truth data, in particular that it may be advantageous to annotating diverse objects to better train a detector instead of annotating the same object across time to create tracking ground truth.

3.2.2 ISBI CELL TRACKING CHALLENGE

In this section we will investigate the performance of nearest neighbour linking (NN) on ISBI CTC data using CP-Net detection. We submitted the combined model, CP-Net+NN, to the ISBI CTC challenge where it was evaluated on two acquisitions for each of the 19 available datasets, and we received DET, SEG and TRA scores for each dataset as well as a ranking for each (dataset, metric) pair.

We view this model as a naive “minimal viable tracker” which optimizes for speed and simplicity. It attempts to reduce the number of unintuitive or opaque hyperparameters to a minimum, choosing instead models with parameters that can be learned efficiently from ground truth annotations (using gradient based optimization) or which are obvious enough that they can be chosen by eye, e.g. *characteristic cell size*, which allows us to avoid brute force hyperparameter optimization.

METHOD DETAILS

The method

1. Rescale XY[Z] spatial dimensions (anisotropically).
2. Normalize raw image intensity via affine rescaling the 2nd and 99.5th histogram percentiles to 0,1 respectively.
3. Produce a sparse, noise free image of approximate cell centerpoint locations with CP-Net .
4. Extract precise centerpoint locations using local peak finding with spherical non-maximum-suppression.
5. Repeat steps 1..4 for every timepoint.
6. Track cells by linking centerpoints with nearest neighbour linking.

Dataset	Rank #1	KIT-Sch-GE	KTH-SE-1	UP-PT	CP-Net+NN
BF-C2DL-HSC	0.978	0.929	—	—	0.975
BF-C2DL-MuSC	0.971	0.967	—	—	0.911
DIC-C2DH-HeLa	0.955	—	—	0.578	0.662
Fluo-C2DL-MSC	0.873	—	0.839	0.478	0.250
Fluo-C3DH-H157	0.987	0.948	0.987	0.931	0.968
Fluo-C3DL-MDA231	0.882	0.820	0.882	0.769	0.747
Fluo-N2DH-GOWT1	0.979	0.949	0.976	0.875	0.902
Fluo-N2DL-HeLa	0.991	0.989	0.991	0.972	0.982
Fluo-N3DH-CE	0.945	0.886	0.945	0.542	0.986
Fluo-N3DH-CHO	0.953	0.948	0.953	0.820	0.878
Fluo-N3DL-DRO	0.785	—	—	0.325	0.785
Fluo-N3DL-TRIC	0.942	0.809	—	—	0.951
Fluo-N3DL-TRIF	0.886	—	—	—	0.954
PhC-C2DH-U373	0.982	—	—	0.883	0.809
PhC-C2DL-PSC	0.966	—	0.959	0.898	0.932
Fluo-N2DH-SIM+	0.975	0.945	0.957	0.896	0.842
Fluo-N3DH-SIM+	0.967	0.933	0.950	0.802	0.907

TABLE 3.2 Rank #1 shows state of the art TRA score on ISBI CTC as of Dec 2021. We compare our TRA scores with the CP-Net+NN model along with the top three performers as ranked by sum total of TRA scores across all datasets. Method scores equal to or greater than Rank #1 are in bold.

For a full description CP-Net architecture and optimization see section 2.3.

The nearest-neighbour tracking is parameter free. Each detection resulting from CP-Net is taken as-is, and linked to it’s nearest parent object in the preceding frame using a Euclidean distance metric that takes into account anisotropic voxel sizes in 3D data.

RESULTS

A high level table of results from this experiment is shown in table 3.2 which includes our TRA scores for each dataset as well as scores from competitors. The full performance data for every competing method is publicly available, which allows us to compare head to head against the top three tracking methods (KIT-Sch-GE, KTH-SE (1), and UP-PT) as ranked by the sum total of TRA scores across all datasets. Additionally we include “Rank #1”, the current top TRA score on each dataset as the state of the art.

In general, the performance of our method, CP-Net+NN, is highly variable across datasets, ranging from state of the art (Fluo-N3DH-CE, Fluo-

N3DL-DRO, Fluo-N3DL-TRIC, Fluo-N3DL-TRIF), to relative failure (DIC-C2D-HeLa, Fluo-C2DL-MSD). In fig. 3.2 we show the relationship between the per-acquisition DET score (there are two acquisitions per dataset) and the number of centerpoint annotations that were available for training the CP-Net model. Clearly the number of annotations is a critical factor in determining the DET score, and because TRA is largely determined by DET the number of annotations is critical here as well. The variance in DET and TRA performance across acquisitions also decreases as the number of annotations is increased. DET and TRA obey a strict 1-1 relationship across all datasets with three exceptions: Fluo-N3DL-DRO, Fluo-N3DL-TRIF, and Fluo-N3DL-TRIC. Each of these three dataset contain one acquisition that purposefully focuses on a period during embryonic development when cells are moving rapidly, e.g. germ band extension in the case of the drosophila dataset, Fluo-N3DL-DRO, and dorsal closure including the elimination of the amnioserosa in the case of Fluo-N3DL-TRIF and Fluo-N3DL-TRIC. The rapid movement appears to make tracking cells much more difficult and is reflected in a significant decrease in TRA scores in the one acquisition with faster moving cells.

CP-Net+NN is an interesting baseline for the tracking performance that is possible when we don't use any ground truth data from outside the ISBI CTC for training and when we use the most naive linking possible. CP-Net+NN does not use the linking step to find missing detections, look for gaps in the cell lineage, or identify false positives. This method has no optical flow or any heuristics to deal with coordinated motion or large displacements. There are no constraints on the number of children possible in a division, on the geometrical relationships between parent and daughter cells, or on the time allowed between successive divisions.

Contrast with other algorithms

Here we cover the details of linking for the three methods KIT-Sch-GE, KTH-SE-1, and UP-PT. Each of these three methods use a detect-then-link approach to cell tracking. Their approaches to detection have already been detailed in section 2.6. KTH-SE-1 achieves state of the art overall tracking scores on multiple datasets by combining a heuristic, filter-based detector containing many tuneable parameters with a linker that iterates the Viterbi algorithm to add tracklets to the solution greedily.

The other methods use some kind of hyperparameter optimization, often brute force search. KIT-Sch-GE uses deep learning for segmentation based on distance-to-boundary regression and trains on external full instance segmentation data. For tracking they implement a modified version of Padfield et al., 2011 that formulates the tracking on pairs of images as a "coupled" min-cost flow problem which can be solved via Lin-

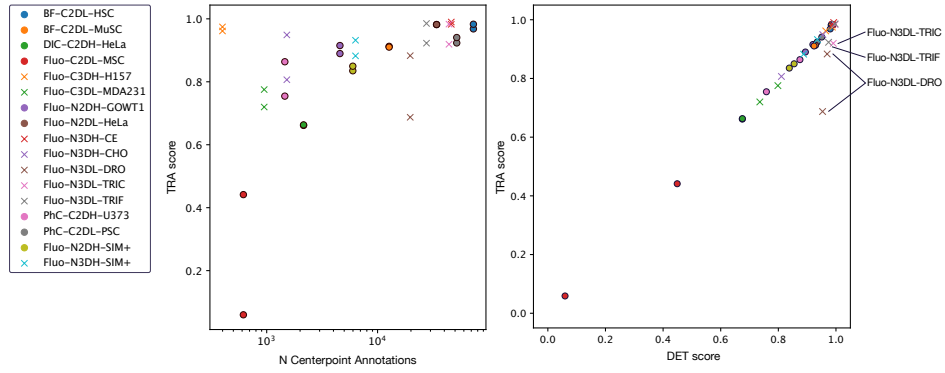


FIGURE 3.2 The number of annotations available for training CP-Net has a strong impact on detection performance (left), which in turn largely determines tracking performance (right). Each dataset has two associated acquisitions and we plot their scores separately. Interestingly, three of the datasets (Fluo-N₃DL-DRO, Fluo-N₃DL-TRIF, and Fluo-N₃DL-TRIC) feature one acquisition with rapid, directed cell movements arising during gastrulation or germ band extension, and a second acquisition with slower movements but more cell divisions. In each case the rapid cell movements cause a substantial decrease in tracking performance, both relative to DET scores as well as to TRA scores on the other acquisition with slow, undirected cell movements.

ear Programming UP-PT uses LoG filter + thresholding with 14 parameters total.

Conclusions Despite these failure cases, our approach is to our knowledge the first which has attempted to provide a tracking solution for all datasets. While it may be unsuitable for some data in its current form, we believe this method potentially provides a simple basis on which to build a truly general purpose cell tracking algorithm.

We believe our method occupies a unique position, a sweet spot, in the space of methods that balances accuracy with broad applicability and simplicity of application. There are several reasons, which, when taken as a whole, make this point. First, the raw input is processed exclusively by a convolutional neural network, which is capable of learning low level features appropriate for each microscope modality featured in the challenge including fluorescence, phase contrast, differential interference contrast, simulated data and bright field. Second, CP-Net only requires (potentially sparse) centerpoint dot annotations for training which are included with each dataset. Third, there are a very small number of hyperparameters, each of which can be chosen either by visual inspection of the raw data, e.g. *average-size-of-cells*, or extracted from the metadata, e.g. *voxel-anisotropy* and the *sparse-annotations* flag. Finally, the inference time of the method is fast, with the forward pass of the CNN on the GPU as the rate limiting step, usually in our experiments at about 10Mpix/sec³. NN tracking can be streamed on pairs of frames as CNN prediction oc-

³training requires backpropagation and runs at 1 2 Mpix/sec

curs on consecutive timepoints, and at normal cell densities has no effect on the marginal processing rate. This means there is still some room for more computation (especially during linking) to improve accuracy while maintaining high throughput.

The tracking results from this experiment vary significantly in quality. While the quantity and quality of CP-Net training data clearly plays a role in determining tracking quality, there is also a role for cell shape (small and round is best) and type of microscopy (fluorescence microscopy is best). More sophisticated linking models can often bridge gaps in detections or correct for individual false negative or positive detections, thereby improving detection accuracy. As nearest neighbour linking does none of these things, we expect it to pay a price not just in tracking accuracy (TRA) but also in detection accuracy (DET). But when we combine a general purpose cell detector CP-Net with fast and simple to implement nearest neighbour linking we get a model that can be evaluated fairly across a broad range of datasets, allowing it to serve as a baseline.

3.2.3 EXPERIMENT - PRETRAINED MODEL ON TRIBOLIUM

In fig. 3.3 we briefly demonstrate the ability of pretrained CP-Net model to generalize to new acquisition data from the same organism. We were able to use CP-Net pretrained on the Fluo-N3DL-TRIF dataset to track H2B labeled nuclei on new, in-house light sheet microscopy data of developing *T. castaneum* directly without any further training / fine-tuning or hyperparameter optimization.

3.3 TRACKING WITH VELOCITY GRADIENT COSTS

In this section we introduce novel “velocity gradient costs” to the Tracking By Assignment framework which improve performance in the presence of large, spatially coherent movements, e.g. muscular contractions or tissue flow during embryogenesis. We implement this model and demonstrate a significant improvement in tracking fluorescence images of nuclei in the planarian worm *Schmidtea mediterranea* which suffers from intermittent muscular twitching. Additionally, we show how velocity gradient costs enable tracking of clustered motile breast cancer cells at significantly reduced framerates.

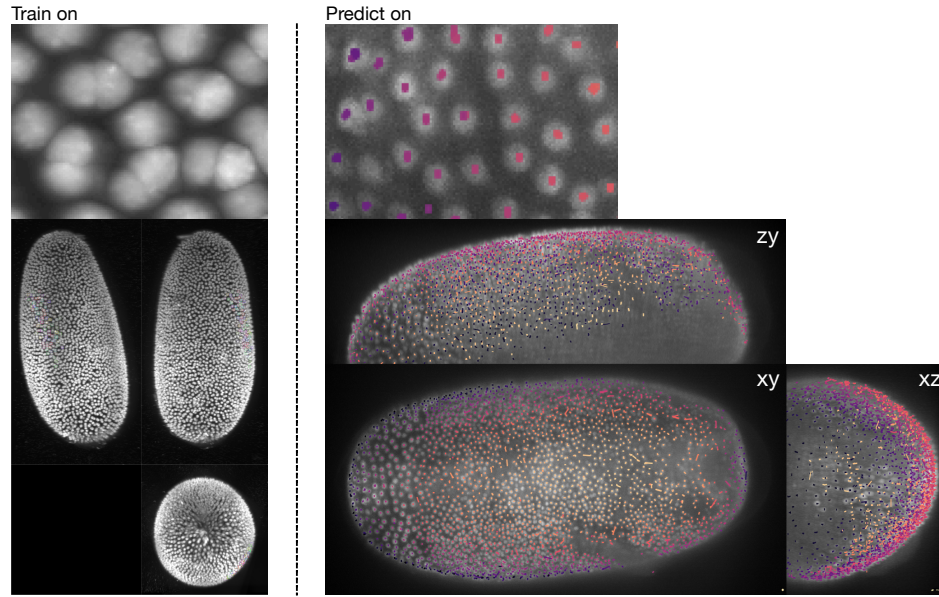


FIGURE 3.3 Here we test a model trained on ISBI dataset Fluo-N3DH-TRIF and applied to in-house data of *T. castaneum* embryogenesis without any finetuning or adjusting of hyperparameters. While the nuclei have generally similar appearance to the human eye, and both datasets were acquired with lightsheet microscopes, the image quality differs significantly. This shows that at least in one case CP-Net is capable of generalizing to a new dataset that has very different image statistics.

3.3.1 MOTIVATION AND RELATED WORK

Tracking by Assignment (TbA) from Kausler et al., 2012 is a popular model for cell tracking made widely available through the Ilastik software (Sommer et al., 2011; Berg et al., 2019).

In standard TbA we first attempt to detect a superset of possible objects in each frame of the timeseries. Then we build a *Chain graph* (Frydenberg, 1990) with a node for each potential detection and a directed edge between pairs of nodes in subsequent frames which capture a superset of potential temporal relationships, i.e. edges describe possible cell movements. Each node and edge is assigned a binary hypothesis variable and a cost depending on its likelihood given the data, and we obtain a tracking by keeping only a valid subset of these nodes and edges which minimizes the sum of costs and does not violate any of a large set of constraints, e.g. a cell division must result in exactly two daughters. This problem can be formulated as an Integer Linear Program (ILP), an NP-Hard but well studied problem in mathematical optimization, and solved to global optimality using widely available optimization software such as Gurobi Gurobi Optimization, LLC, 2022, CPLEX (Cplex, 2009) or COIN-OR (Lougee-Heimer, 2003). Recently fast approximate solvers have been proposed (Haller et al., 2020) specifically targeting the cell-

tracking with TbA models reducing computation time and memory, and enabling the solution of larger tracking problems.

The separation of the detection and linking phases, and the abstract formulation in terms of a chain graph makes the TbA model highly flexible, and multiple variation on the original model have appeared introducing new assignment variables, costs or constraints specifically adapted to unique datasets. For instance Schiegg et al., 2013 enables tracking undersegmented clusters of cells while ensuring that the number of cells in each cluster is properly accounted for. Jug et al., 2014 includes prior knowledge of the specific geometry of the experimental setup as a constraint to maintain the spatial order of cells in a narrow, cylindrical capillary while allowing for cell divisions. Jug et al., 2016 enables joint segmentation and tracking by introducing a “morality” constraint on neighbouring superpixels which removes invalid cell merges from the solution space.

But TbA struggles when cell displacements are very large relative to the inter-cell spacing and there is little variation in cellular appearance. In this context, the matching may be ambiguous and the problem ill-posed without a regularizer on the displacement field. Approaches which naturally incorporate such regularizers e.g. point-set matching or non-rigid image registration techniques may seem a better fit. Non-rigid image registration methods easily incorporate smoothness in the form of regularizers on the gradient or Laplacian of the continuous displacement field (Cahill et al., 2009a; Cahill et al., 2009b; Thirion, 1998). Also, two-phase algorithms which optimize, alternately, a smooth, non-rigid spatial transformation followed by a discrete matching have been used in atlas-based registration. E.g. Qu et al., 2011 use thin plate splines to enforce spatially smooth displacement fields before performing a discrete matching. Also, Kainmueller et al., 2014 perform atlas-registration using two-phase optimization of an affine transformation followed by a discrete matching energy very similar to ours which includes a cost for local pairwise differences in the displacement of neighbours. But thus far smooth displacement fields have not been incorporated directly into the tracking by assignment framework.

The typical solution in the context of cell tracking is to include a separate optical flow computation on pixels or superpixels prior to linking and assignment of object relationships. Blasse uses a technique from computer vision Sun et al., 2010 to compute optical flow of cell movements during hinge contraction in the wing epithelium of *D. melanogaster* (Blasse, 2016). Also, Amat develops a custom approach to optical flow computation designed for whole-embryo tracking during early development and gastrulation (Amat et al., 2013).

Once optical flow has been computed, one can then use the dense flow field to predict the position of each object projected into the next frame. The positions of objects at time t can be extrapolated to $t + 1$ and then the standard, distance based assignment costs can be computed from the new positions. The advantage of the optical flow based approach is that patterns in the image intensity which do not correspond to detected objects, e.g. dead cells or extracellular material, can still be used to infer the overall flow field, and incorporating smoothness regularization is straightforward.

But typically, optical flow lacks any biological knowledge about the objects in the field of view, which can lead to problems as cells undergo conformational changes throughout the cell cycle, as well as after cell divisions or apoptosis. TbA explicitly models these biological events in the assignment costs, whereas optical flow must rely on image similarity measures like the l_p -norm of intensity differences or cross-correlation Lucas and Kanade, 1981; Willert and Gharib, 1991.

3.3.2 MODEL

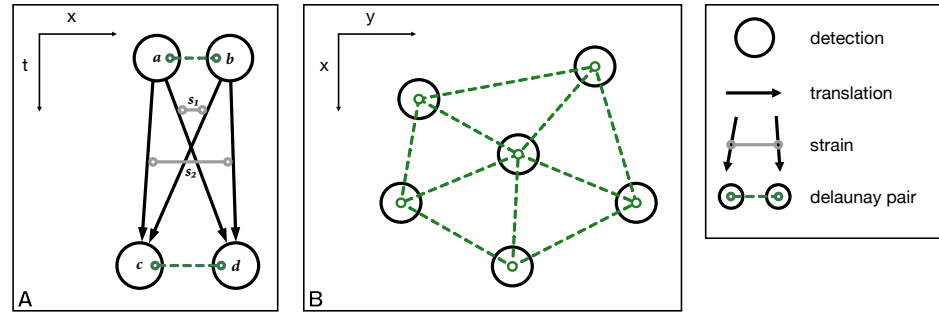


FIGURE 3.4 For each timepoint, we add binary velocity gradient (VG) variables to neighbouring Delaunay pairs of translation vectors and associate a cost proportional to the square of the relative displacement. For example the velocity gradient s_2 carries a cost proportional to $\|(r_c - r_a) - (r_d - r_b)\|_2^2$ for the detections a, b, c, d . This cost models a simple spring potential between each pair of neighbours and penalizes non-rigid warping of the domain, encouraging smooth flow fields appropriate for e.g. densely packed cells in a growing or flowing tissue. **(A)** Grey “velocity gradient” variables s_1, s_2 connect pairs of translation vectors between detection centerpoints across time. Each VG variable s has an associated binary variable x_s and cost c_s in the ILP. **(B)** Green edges show detections which are neighbours in the Delaunay triangulation of detection centroids. For each Delaunay pair we add VG edges to all pairwise combinations of translation vectors (black, directed edges).

Here we propose a technique that allows TbA to incorporate a smoothness prior on the discrete displacement field describing cell movements, obviating the need for optical flow. The regularized model is solved in the standard way by optimizing a single, global integer linear program (ILP)

without resorting to iterative, two-step optimization. To do this we introduce binary “velocity gradient” variables for each pair of neighbouring translation hypotheses. These variables carry costs which only penalize differences in the displacements of neighbouring objects and are therefore invariant to rigid transformations of centroid positions, i.e. translation and rotation.

The intention behind adding a cost to differences in displacement is to prefer tracking solutions where neighboring cells move in the same direction. We can justify the costs on statistical grounds whenever we observe locally correlated motion, but the cause of correlated cell movement is often physical contact. Cells exert physical forces on their neighbors through tight junctions in tissues, through bacterial pilus, or collisions between motile cells.

We implement velocity gradient costs as proportional to the magnitude of the difference between neighbouring displacement vectors, e.g. $\text{cost}(a \rightarrow c, b \rightarrow d) = g(\|(\mathbf{x}_c - \mathbf{x}_a) - (\mathbf{x}_d - \mathbf{x}_b)\|_2)$ (see Figure fig. 3.4). We add the velocity gradient variables and costs to the standard tracking by assignment linear objective. It is also necessary to introduce a set of constraints to ensure that velocity gradient variables are consistent with their associated translation variables. Note that divisions are excluded from this cost. The combined system is then optimized using Gurobi (Gurobi Optimization, LLC, 2022), a standard ILP solver.

FORMAL MODEL DESCRIPTION

First we define the set of all detections $\mathcal{D}$ and for every detection $d \in \mathcal{D}$ we have time τ_d and location r_d . Then we can define translations $\mathcal{T}$ and velocity gradients $\mathcal{S}$ as

$$\mathcal{T} = \{(a, b) \in \mathcal{D}^2 \mid a \in \mathcal{D}_\tau, b \in \mathcal{D}_{\tau+1}, |r_a - r_b| < d_{\max}\} \quad (3.2)$$

$$\mathcal{S} = \{((a, b), (c, d)) \in \mathcal{T}^2 \mid (a, c) \in \text{delaunay}(\mathcal{D}_\tau) \text{ and } b \neq d\} \quad (3.3)$$

where $d_{\max}$ is a data-specific distance cutoff, and $\text{delaunay}(\mathcal{D}_\tau)$ is the set of edges in the Delaunay graph of all detection centerpoints at time τ (dashed green edges in fig. 3.4). To each detection d , translation t and velocity gradient s we assign a binary variable, x_d , x_t and x_s respectively, that takes a value $x = 1$ if the object is *active*, i.e. a part of the tracking solution, and zero otherwise.

Next, we assign a fixed cost to each binary variable. In our implementation, the functional form of these costs can be given at runtime, making it

easy to customize costs per dataset. However, we typically use a cost for a translation $t := (a_1, a_2)$ proportional to the Euclidean distance between associated detections, $c_t = \|r_{a_1} - r_{a_2}\|_2/d_0 - 1$ where d_0 is the distance beyond which $c_t > 0$ and thus discouraged. The cost associated with velocity gradient variables $s := ((a_1, a_2), (b_1, b_2))$, for all experiments is $c_s = \|(r_{a_2} - r_{a_1}) - (r_{b_2} - r_{b_1})\|_2/d_0 - 1$, where d_0 is the distance beyond which $c_s > 0$ and thus discouraged. The result is proportional to the magnitude of the difference between neighboring translation vectors. In situations where our detection method cannot provide a meaningful cost for each detection we use a fixed cost, $c_d = 0$.

The objective is then a dot product between a collection of binary variables and their costs,

$$f(\mathbf{x}) = \mathbf{x} \cdot \mathbf{c} = \sum_{\ell \in \mathcal{D} \cup \mathcal{T} \cup \mathcal{S}} x_\ell c_\ell \quad .$$

Finally, we introduce a set of constraints which serves two purposes. First, they maintain internal consistency between related sets of detection, translation and velocity gradient variables. And second, they allow us to introduce biologically meaningful restriction on the structure of our tracking solution. The individual constraints introduced are

$$x_d \geq \sum_{t \in \text{in}(d)} x_t \quad \forall d \in \mathcal{D} \quad (3.4)$$

$$2x_d \geq \sum_{t \in \text{out}(d)} x_t \quad \forall d \in \mathcal{D} \quad (3.5)$$

$$x_{t_1} + x_{t_2} \geq 2x_s \quad \forall s = (t_1, t_2) \in \mathcal{S} \quad (3.6)$$

$$x_{t_1} + x_{t_2} \leq 1 + x_s \quad \forall s = (t_1, t_2) \in \mathcal{S} \quad (3.7)$$

where $\text{out}(a) := \{(a, b) \in \mathcal{T}\}$ and $\text{in}(a) := \{(b, a) \in \mathcal{T}\}$. These mathematical constraints have the following implications:

- cells do not merge [eq. (3.4)]
- cells do not divide into more than two daughters [eq. (3.5)]
- cells may appear spontaneously⁴ [eq. (3.4)]
- cells may disappear (apoptosis) [eq. (3.5)]

⁴This constraint is not always desirable, e.g. when we know cells do not enter the field of view through image boundaries. Replacing $\geq$ with $=$ in eq. (3.4) accomplishes this without compromising other properties.

- velocity gradient variables are active if and only if both translations are active [eq. (3.6) and eq. (3.7)].

3.3.3 EXPERIMENTS - TWITCHY PLANARIAN

We apply this model to an image time series which naturally features large, nonrigid warping between frames. “Large” in this context means “at least as large as half the average distance between neighboring cells”, which is when nearest neighbour tracking begins to fail. The planarian worm *S. mediterranea* is a new model species for studying regeneration whose natural transparency promises to simplify image based studies of regeneration. Our collaborator Tobias Boothe acquired a timeseries of H2B-GFB *S. mediterranea* post lesion to study its regeneration in terms of the migration of single cells. However, unfortunately, each specimen exhibited muscular twitches in response to high-powered illumination in the wavelength of normal fluorescent probes. This twitching movement is much faster than the time between stacks, resulting in large displacements between frames but no motion blur. Further, the H2B labeled nuclei are uniform in appearance, which together with large displacements makes tracking difficult. However, we were able to use the VG cost model to help our tracker correctly follow cells despite these large, nonrigid deformations.

In fig. 3.5 we evaluate our approach we hand annotated the trajectories of a sparse subset of 33 nuclei equally spaced throughout the tissue over eight frames of the timeseries which included three $(t, t + 1)$ frame pairs (1,2,4) separated by large cell movements. We use three different linking methods on an identical set of 3112 total noisy cell detections obtained by the standard RandomForest based cell detection pipeline of the ilastik software Sommer et al., 2011. We detect cells in 2D max-projection using a small pixelwise classifier from ilastik, and use this fixed set of detections to compare three alternative linking methods. Due to our sparse annotations we use the “link recall” metric ($|TP|/|T|$) on the set of edges relating objects between times t and $t + 1$. While each method performs equally well on frame pairs with small displacements, the addition of VG costs the the TbA framework allows it to make the correct links even for frames where the displacement is large and smooth.

One downside to the VG-cost approach is a large increase in runtime due to the additional VG-variables and constraints. The number of VG variables added to the system depends primarily on the number of detection variables $|\mathcal{D}|$ and the number of $t + 1$ neighbours k we include when creating translation hypotheses. A useful approximation is $3|\mathcal{D}|k^2$ where the factor 3 comes from the average number of Delaunay edges per cell

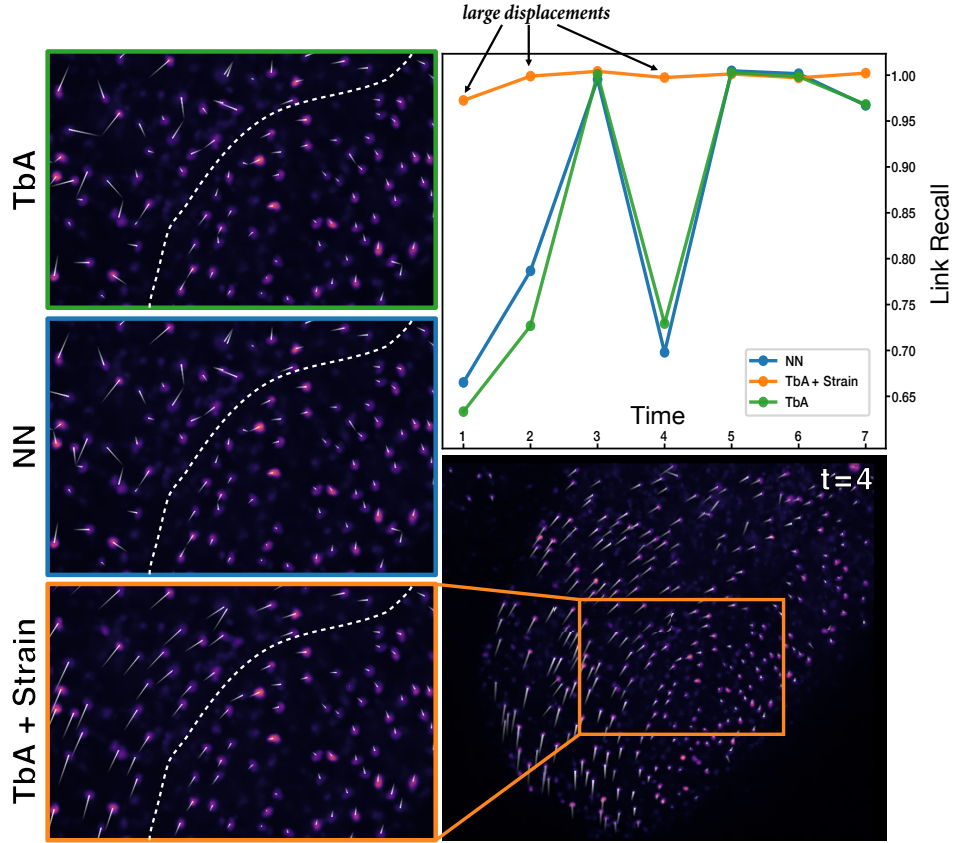


FIGURE 3.5 Velocity gradient (VG) costs help tracking across large displacements. Plot shows link recall for three tracking methods on a short timeseries of fluorescent nuclei from regenerating *Shmidtea mediterranea*. (bottom right) A single frame ($t = 4$) with cell displacement vectors obtained by TbA+VG immediately after muscular twitch. (Left column) compares the links generated by the three tracking methods for $t = 4$. While all methods obtain similar movements on the region of tissue with small displacements (right of white dashed line) TbA and NN are unable to find the true, smooth flow when in the region of tissue with larger displacements (left of white dashed line).

and the factor k^2 comes from selecting all pairs of translation hypotheses. In practice we found that nearest-neighbour tracking required $190 \text{ ms} \pm 11.4 \text{ ms}$, while Gurobi branch-and-cut ILP solver required TbA needed $5.37 \text{ s} \pm 291 \text{ ms}$, and TbA with VG costs needed $11 \text{ min } 54 \text{ s}$ and $14 \text{ min } 44 \text{ s}$ for solver run on $(t, t + 1)$ -pairs and joint solver, respectively. The two VG-cost solver approaches produced nearly identical solutions. fig. 3.5 shows the results using solve-then-fuse over $(t, t + 1)$ -pairs for TbA with VG costs.

3.3.3.4 EXPERIMENTS - BREAST CANCER CELLS IN DISH

We also evaluate the VG cost tracking model on human breast cancer cells in fig. 3.6, simulating data of lower temporal resolution by subsampling frames in time. The motions of cells are correlated despite not being

in tissue, but from simple steric hindrance. The dataset is composed of 23070 unique detections across 86 timepoints, and both NN and TbA + VG linking models use the same set of detections. In it's original form the data has high temporal resolution and small cell displacements between frames, which allows nearest neighbour tracking to work well. But by removing every n -th frame we simulate data with lower temporal resolution and larger displacements, causing NN tracking to fail. But by using TbA with VG costs we can maintain correct cell links over much larger displacements at lower temporal resolution.

The cells enter the field of view through the top image border and flow downwards. It could be that the motions of the cells are well described by a random walk superimposed on a constant velocity downward flow. To see if this is the case we also include two additional linking models which supplement NN and TbA+VG with an initial translation of detection centerpoints to align the centers of mass at frames t and $t + 1$ before linking them. Indeed this improves NN significantly by helping to compensate for the constant downward flow, but despite this addition it is still outperformed by VG cost tracking both with and without this initial registration.

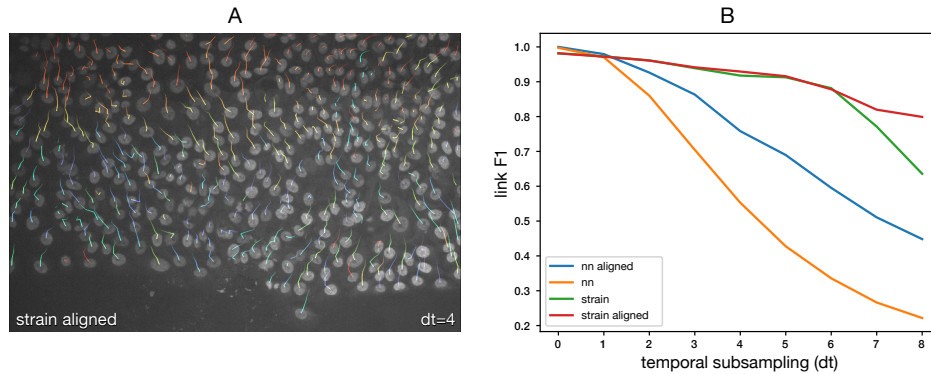


FIGURE 3.6 Velocity gradient (VG) costs allow us to track at reduced frame rates. Here a timeseries of densely packed breast cancer cell in a dish move in a constrained environment. While the high framerate allows nearest neighbour tracking to work initially, we can simulate a reduced framerate by subsampling this dataset in time, causing nearest neighbour tracking to fail. VG cost tracking however manages to maintain high accuracy even for large subsampling and longer displacements between cells. **(A)** Breast Cancer cells in a dish with tracking “tails” highlighting the smooth flow fields outlined by the cells’ motion. The tails shown are the result of tracking with VG costs and center of mass alignment (VG aligned) on 4x subsampled data. **(B)** A comparison between VG cost tracking and simple nearest parent linking with/without center of mass registration. For comparison we evaluate the F1 score ($2|TP|/(|T| + |P|)$) on links using a pseudo ground truth created by nearest neighbour tracking on the dataset without any subsampling. At high framerate (low subsampling) all methods achieve solutions similar to our pseudo Ground Truth of nearest-neighbour tracking on data without subsampling. Increasing the displacement between frames cannot be corrected by center of mass alignment alone, but is significantly improved by our VG cost model which prefers locally smooth flow fields.

3.4 DISCUSSION

In this chapter we have shown a state of the art cell tracking model based on highly accurate detections provided by CP-Net . Using this tracking model we were able to cut the error rate significantly on 3/4 developmental dataset in the ISBI Cell Tracking Challenge while using only a fast, parameter-free nearest neighbour linking model. In the second half of this chapter we explored a particular failure mode of naive nearest-neighbour linking when the displacements of cells becomes large relative to their nearest-neighbour distance. We then introduce the VG cost tracking model and show how it can restore accurate linking in cases where these large displacements form smooth flow fields. In future work we aim to address the major shortcoming of the VG cost approach, runtime, through the introduction of fast heuristics for optimization.

REMOVING STRUCTURED NOISE WITH SELF-SUPERVISED BLIND-SPOT NETWORKS

Computational denoising is an important first step in many biological image analysis pipelines. Fluorescence bioimages in particular often suffer from noisy and grainy textures exacerbated by the desire to minimize light exposure to the sample. This noise hides the content of the image, making both visual and automated analysis more difficult. Progress in convolutional neural networks over the last few years has led to an explosion of image restoration methods for removing noise and improving image quality, but current state-of-the-art network approaches require expensive-to-collect, clean ground-truth images. While recent *blind-spot* based network training methods have removed this requirement for specific cases, they remain unable to remove the *structured noise* common to some kinds of imaging hardware. In this chapter we present *Structured Noise2Void* (STRUCTN2V), a generalization of the blind-spot approach which enables the removal of structured noise without resorting to the use of ground-truth data or an explicit noise model. Specifically, we propose a specialized network training scheme making use of an extended *blind mask* (rather than a single pixel/blind spot), whose shape is adapted to the structure of the noise. We evaluate this approach on a variety of real fluorescence microscopy datasets and show that STRUCTN2V considerably improves the removal of structured noise compared to existing standard and blind-spot based techniques.

4.1 MOTIVATION

One of the major issues which inhibits our ability to obtain the maximum information from microscopy images is the corrupting presence of image noise. Many of the hardware and sample preparation decisions made, (e.g. what concentration of the label/marker we need, whether to actively cool the image sensor) are made to minimize the effect of noise or maximize the relative strength of the signal. Despite these efforts, acquisition noise of some kind is common to all microscopes and often significantly hampers our ability to make sense of their output. Thus,

purely computational image denoising is a fundamental part of almost all bioimage analysis workflows. When performed successfully it facilitates both the visual understanding of images as well as any downstream processing like detection, segmentation or tracking (Ulman et al., 2017; Weigert et al., 2018; Wu et al., 2008). This is especially true for live cell imaging, where a trade-off often exists between the signal-to-noise ratio of our image and the health of our sample or specimen and thus the validity of our experiment. But removing noise from bioimages can be quite difficult, and there is as of yet no one-size-fits-all, general purpose solution which works well across the full spectrum of biological samples and imaging hardware.

4.2 RELATED WORK

There are a wide variety of existing approaches which vary greatly in accuracy, generality, speed and efficiency. Simple low-pass filtering may be sufficient if our data is already very high quality, especially when the downstream processing is robust against noise e.g. Amat et al. (2014) and Stegmaier et al. (2016). However when the noise is significant, many biologists and bioimage analysts apply well established, general purpose techniques like NLM (Buades et al., 2005) or BM3D (Dabov et al., 2006). These methods are usually straightforward to apply, require minimal hyperparameter tuning, and are sometimes even integrated into commonly used tools such as Fiji (Schindelin et al., 2012), scikit-image (van der Walt et al., 2014) or gputools (Weigert, 2019). However these approaches still leave much room for improvement both in terms of accuracy and speed. One generic way to achieve better accuracy is to use a method with a more narrow scope (and more built-in prior knowledge) that better fits the data in question, e.g. a method designed for sCMOS noise removal if your microscope has an sCMOS detector (Mandracchia et al., 2020). However, there is a much wider variety of hardware and image types than there are denoising methods, so searching for an appropriate method requires time consuming additional research which may end up fruitless. And even if you find a more specific method for your setup, it may still be insufficiently accurate, and is likely to have a less efficient (or even less correct) implementation than one of the popular, generic denoising methods.

An alternative strategy relies on learning from the data and usually has relatively little built-in prior knowledge about the hardware or the biological sample ¹. Methods based on learning usually result in denois-

¹While in theory methods could exist that have both lots of built-in prior knowledge and lots of learning ability, this domain is not well explored. This is a result of the Machine Learning field's emphasis on increasing flexibility/power of learners, as well as the practical challenge of integrating the two approaches. Stardist

ers highly tuned for the data seen during training, but can be capable of excellent performance without requiring a lot of application-specific, hand-coded prior knowledge and engineering.

Starting around 2005, learned methods based on Random Fields such as Fields of Experts (Roth and Black, 2005), Shrinkage Fields (Schmidt and Roth, 2014) surpassed engineered approaches in terms of accuracy while maintaining reasonable performance. This trend has only continued as deep learning and especially Convolutional Neural Networks (CNNs) replaced approaches based on Random Fields improving representational power and inference time (K. Zhang et al., 2017; J. Xie et al., 2012; Vincent et al., 2010; Jain and Seung, 2009). The inherent modeling power of CNNs has enabled them to outperform heuristic and Random Field based approaches, while adding the benefit of the fast inference (thanks to modern GPUs) necessary for modern digital microscopy. The largest remaining cost comes during training time, which is longer and more involved than optimizing the few parameters of heuristic approaches or the relatively fast training of Random Field approaches. Within bioimaging these trade-offs make sense, and CNN based denoising approaches have been very successful.

This training phase can vary greatly between CNN-based denoisers, principally in terms of how much *supervision* is required during training. Supervised training requires a special kind of data (matching pairs of noisy and clean/“ground truth” images) that in general requires a specialized acquisition procedure and close control over the microscope. When this data is available supervised methods based on CNNs such as CARE (Weigert, 2019) can provide truly impressive denoising results. The alternative “unsupervised” or “self-supervised” approaches don’t require a special clean version of the image, and instead attempt to learn just from the raw, noisy data. Typically, self-supervised training is performed on some *pretext task* the purpose of which is to teach the network to recognize informative features that will also be useful on the *downstream task*, e.g. image classification, detection or segmentation. Examples of self-supervised pretext tasks include predicting image orientation (Gidaris et al., 2018), re-colorizing greyscale images (R. Zhang et al., 2016; Larsson et al., 2016), and re-sorting images based on spatial or temporal order (Lee et al., 2017; Noroozi and Favaro, 2016; Doersch et al., 2015).

Self-supervised approaches to denoising have also been explored. Recent *blind-spot* networks *Noise2Void* by Krull et al. (2019) and *Noise2Self* by Batson and Royer (2019) have introduced self-supervised training designed explicitly for denoising. Under certain circumstances these approaches

by Schmidt et al. (2018) is an example of the benefits resulting from modifying a very powerful learning architecture to follow constraints (in this case cell shape) that are appropriate for the specific task of nucleus segmentation.

can show performance comparable with CARE without the burden of acquiring GT training data. But they are limited to removal of *pixelwise independent* noise sources, such as most shot noise (typically Poisson), or instrument read noise (typically Gaussian), but struggle when the noise is locally correlated between pixels. Potential causes of such correlations include correlated readout lines on CCD (or even sCMOS) image sensors or correlations along fast-scan lines in some confocal microscopes. In this chapter we capitalize on the recent progress of blind-spot networks for high quality, fully self-supervised denoising and extend it to allow for denoising some classes of structured noise.

4.3 STRUCTN2V METHOD

STRUCTN2V shares much with other recent blind-spot based denoising techniques, and in particular with N2V. Most obvious is that they are self-supervised techniques, i.e. they don't need expensive *clean* (ground truth) copies of the raw data for training. Importantly, removing this burden means that these techniques can be applied by third parties, e.g. bioinformatics facilities who have no access to or control over the microscope hardware. Additionally, these techniques can be applied to *older* data potentially acquired on an out-of-service microscope, or using long-forgotten sample preparation and acquisition parameters.

The most important reason underlying *why these approaches work* is also shared, which is that these techniques remove noise because they are unable to predict it². This is true independent of how the blind-spot is implemented. It is also true of non blind-spot based techniques like Noise2Noise (Lehtinen et al., 2018) and CARE to an extent, although CARE primarily removes noise by training on nearly-noise-free ground truth data. As the noise in the output is uncorrelated with any part of the input the only way to predict it is to memorize it, which can be prevented by the usual means of preventing overfitting (few training epochs, lots of training data). However in self-supervised losses the noise in the output perfectly correlates with the noise in the input, because the images are the same. So we need a clever way to prevent the net from learning the trivial, but optimal identity function. Blind-spot based approaches achieve this by hiding pixels in the input from the network, and then asking the network to predict the hidden value using the surrounding pixels for context.

How do we "hide" pixels from the net? Different techniques take different approaches. Laine et al. (2019) build the blind spot into the architecture of the network directly by hard-wiring in a new shape to the receptive

²in the output, as well as they can predict the signal

field. However we adopt the original, more flexible approach taken by N2V and Noise2Self which replaces the values of pixels in blind spots with random pixel values before passing them on to the network, see fig. 4.1.

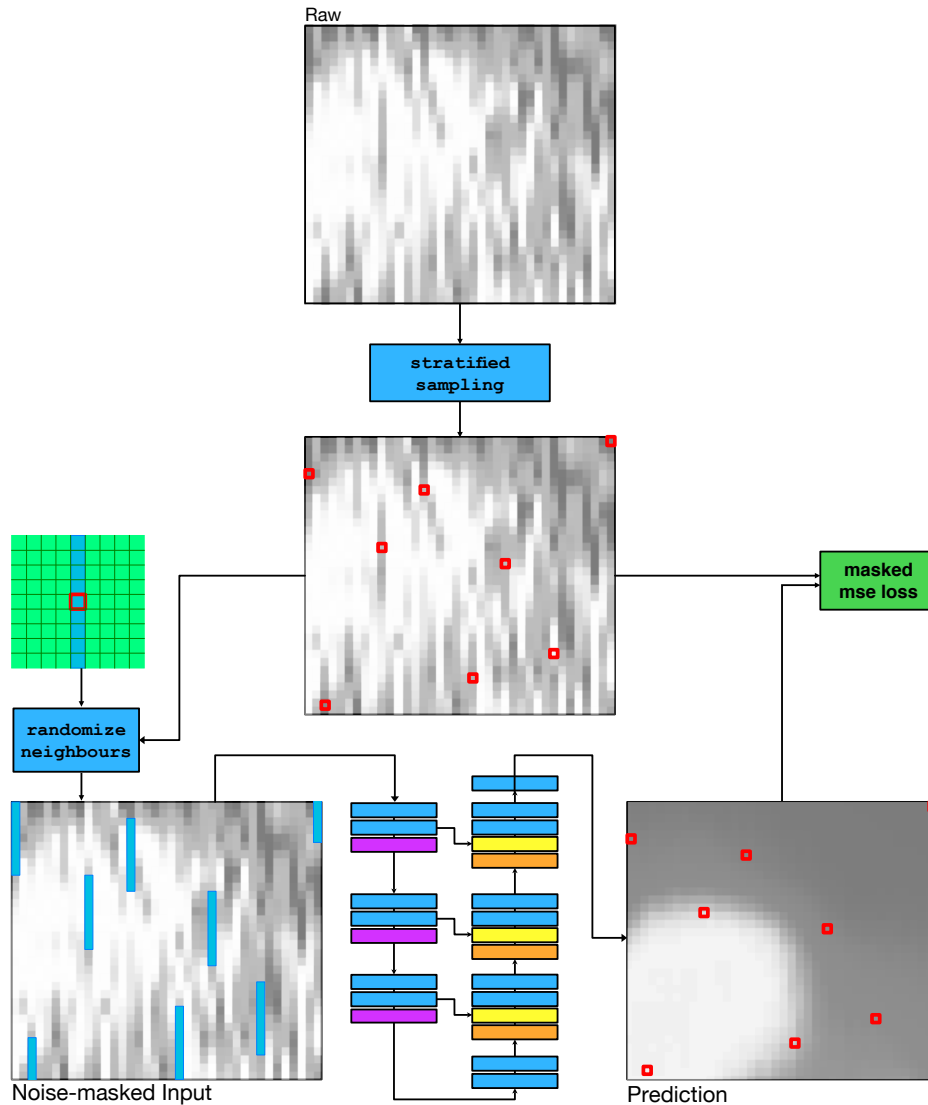


FIGURE 4.1 Overview of STRUCTN2V training. Roughly 2% of pixels within a patch are selected via stratified sampling (red pixels). The values of pixels in the *blind mask* (blue) are replaced with a random value drawn from a flat distribution on $[0, 1]$. This blind mask image is fed to STRUCTN2V model (a variant of U-Net) which predicts a *noise free* image. An MSE loss is computed only on the red pixels typically at the center of the blind mask.

As we are about to see, the flexibility of this approach to “hiding” pixels will also help us overcome the inability of existing methods to remove structured noise. Structured noise is the part of the noise that has *spatial correlations*. This kind of noise is often introduced by the microscope hardware, e.g. lateral stripes introduced by a shared output amplifier noise

in charge-coupled device (CCD) based image sensors citeCCD noise and example from Alex. Existing blind-spot based approaches can use the spatial correlations present in this noise to predict it's value *at the blind spot*, and this predictability prevents it being removed from the output of the network.

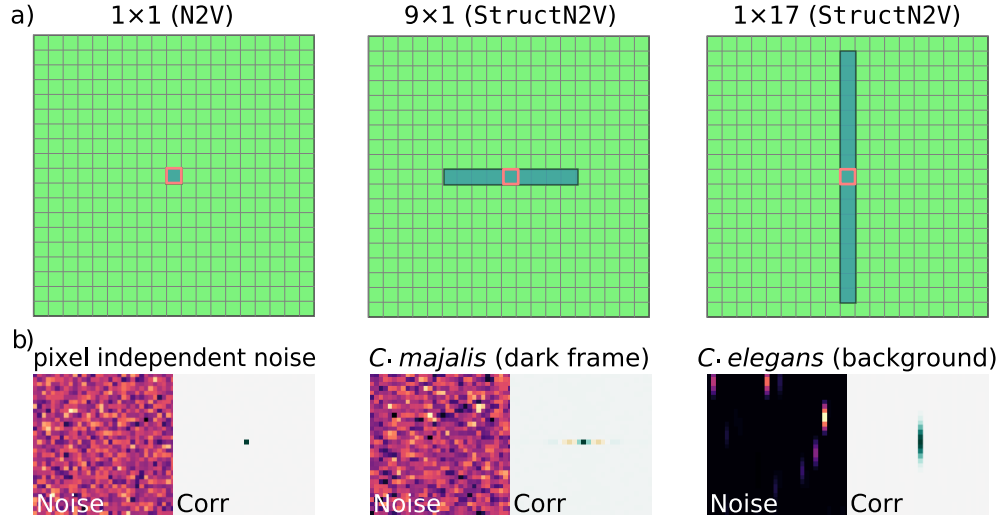


FIGURE 4.2 (a) Masking schemes for N2V (left) and two variants of STRUCTN2V (middle, right). We show the *active pixel* (red), the hidden *blind mask* (blue) and the pixels available (green) to the network to predict the value of the active pixel. (b) Example crops of pure noise images and their spatial autocorrelation (Corr) for pixel independent Gaussian noise (left), a dark frame of dataset *C. majalis* (middle), and a background selection (see fig. 4.4) of dataset *C. elegans* (right). Note that the spatial extent of the autocorrelation pattern in each case suggests the corresponding blind mask depicted directly above in (a).

STRUCTN2V attempts to solve this problem by hiding *additional* pixels from the input (see the blue pixels in fig. 4.2) surrounding each of the target pixels present in the loss (red boxed pixels in fig. 4.2).³ However, this presents a tradeoff, as the blue, randomized pixels are no longer useful for predicting the *signal* either, and so we must try to randomize the pixels which correlate most strongly(weakly) with the noise(signal).

Selecting the blind mask

An appropriately shaped blind mask is critical to achieving optimal results with STRUCTN2V.

Fortunately in many cases it may be possible to estimate the shape of the blind mask by eye from the image itself, e.g. simply by looking at the direction of line artifacts in the raw *C. elegans* data or observing the horizontal stripes in *C. majalis* after initial application of N2V (fig. 4.6c). However, there are more robust approaches which require minimal effort

³Note that the hiding of arbitrary pixels within the receptive field of a given target pixel is not possible with the approach taken by Laine et al. (2019).

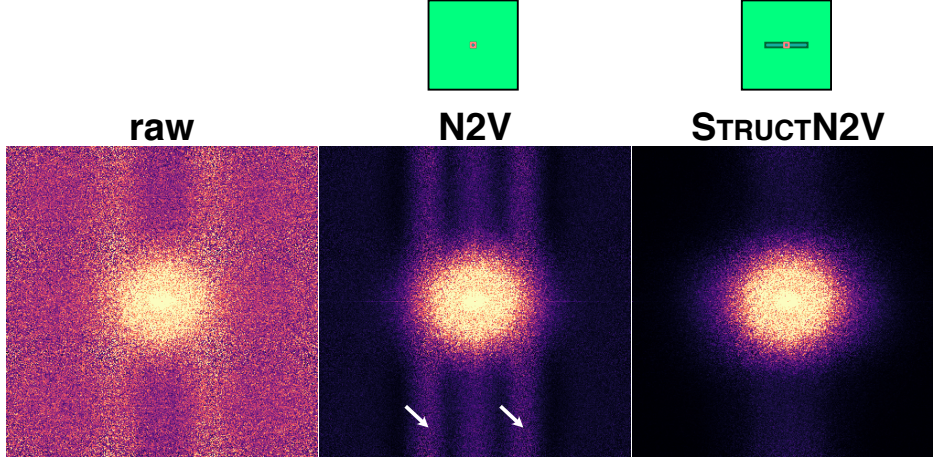


FIGURE 4.3 Comparison of Fourier power spectrum of raw and denoised images. Note the vertical bands present after application of N2V correspond to a pure horizontal signal with characteristic wavelengths of a few pixels. These can be removed by STRUCTN2V without sacrificing the true signal present within a disk centered at the origin. Insets depict the blind masks used during training.

and may help when then correlations in the noise are less obvious. One method which we used to confirm our choice of mask for the *C. elegans* dataset was to compute and plot the *autocorrelation of the noise* (fig. 4.2b (right half)). The autocorrelation of a “pure noise” image clearly reveals the pattern of correlations in the noise and we can choose a mask shape which mostly covers the pixels with large values in this autocorrelation. In fluorescence images like *C. elegans* where the content is only present in the center of the image we can (manually) select and crop out background regions near the edges (fig. 4.4). Similarly we can obtain a pure

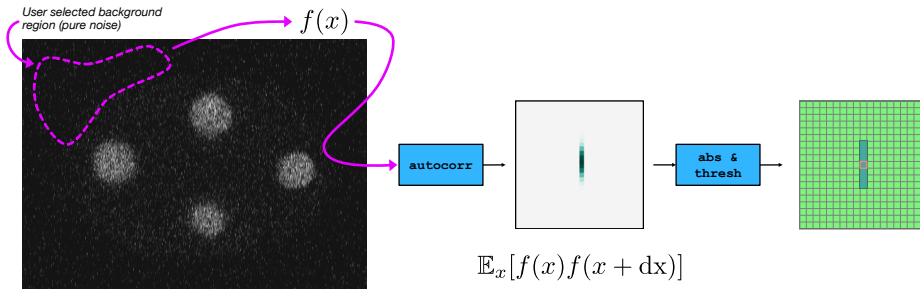


FIGURE 4.4 The user selections a content free region of the image for use in autocorrelation.

noise image by acquiring a “dark frame” with the camera shutter closed (fig. 4.6e) which allows us to observe the read noise of the microscope in the absence of any input signal. Again, the autocorrelation of such an image helps us to see the pattern of correlations in the noise and is useful in estimating the blind mask shape (fig. 4.6f).

An alternative approach is to visualize the image in the frequency domain, which in the case of *C. majalis* was helpful for confirming the rough size of noise correlations *without resorting to an additional dark frame acquisition* (fig. 4.3).

It is important to remember that, in theory, the blind mask shape determines a tradeoff: masking additional pixels prevents the net from learning useful correlations within the signal, but simultaneously prevents the net from learning unwanted correlations within the structured noise. However, despite this tradeoff, there may be a range of mask shapes which are suitable in practice (fig. 4.7 9×1 - 13×1). Finally, using the truly optimal mask shape may be unnecessary so long as the image quality resulting from the denoising procedure is sufficient for all downstream tasks.

4.4 THEORY OF BLIND SPOT BASED DENOISERS

In this section we offer a mathematical explanation for different aspects of the `STRUCTN2V` method. In particular we

- explore the relationship between `STRUCTN2V` and fully supervised methods
- contrast the relationship between `STRUCTN2V` and other blind spot denoising approaches like `N2V`, `Batson` and `Laine`.
- and highlight the tradeoff between reducing noise and improving signal in our choice of blind spot shape.

HOW ARE `STRUCTN2V` AND SUPERVISED TRAINING SIMILAR?

For some raw image $x = y + \eta$, signal y and noise η we aim to find a function which minimizes the self-supervised loss $L = \mathbb{E}(f(x) - x)^2$ over the data distribution $Pr(x)$. and expand the loss as follows,

$$L = \frac{1}{N} \sum_i^N \ell_i \quad \text{where} \quad (4.1)$$

$$\begin{aligned} \ell &= (f(x) - x)^2 \\ &= (f(x) - y - \eta)^2 \\ &= \underbrace{(f(x) - y)^2}_{\text{supervisory signal}} - 2 \underbrace{f(x)\eta}_{\text{model/noise covariance}} + \underbrace{\eta^2 + 2y\eta}_{\text{const}}. \end{aligned} \quad (4.2)$$

Here L is the pixelwise average of ℓ over the image domain, and all operations in ℓ (except for f) apply pixelwise as well.

Under naïve training, the network f learns the identity function, $f^*(x) = \operatorname{argmin}_f L = \mathbb{E}x \mid x = x$, and none of the noise is removed. This can be seen in eq. (4.2) as the presence of the $f(x)\eta$ term⁴ which encourages the network to reproduce the noise. Blind-spot based approaches like `STRUCTN2V`, `N2V`, etc. can be seen as an attempt to restrict the family of functions $\mathcal{F}$ for the network f to only those where

$$\mathbb{E}f(x)\eta = 0 \forall f \in \mathcal{F} \quad , \quad (4.3)$$

i.e. the covariance between the model prediction and the image noise is zero, which they do by making it very difficult for the network to learn the noise directly⁵. The only remaining non-constant term in the pixelwise loss ℓ is the *supervisory signal*, $(f(x) - y)^2$, which is then minimized.

HOW CAN WE MAKE IT DIFFICULT FOR THE NETWORK TO LEARN THE NOISE?

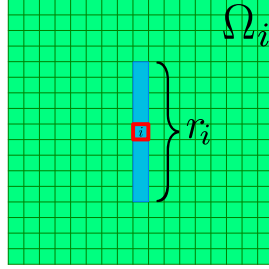


FIGURE 4.5 The barrier region (blue) separates the visible receptive field (green) from the target pixel (red). These regions obey $i \subseteq r_i \subset \Omega_i$. We can use this to define the visible receptive field, $v_i := \Omega_i \setminus r_i$.

Blind spot based techniques essentially make the noise “invisible” to the network. Let us consider the perspective of a single pixel $x_i = y_i + \eta_i$ which the network is trying to predict (see fig. 4.5). The pixel i is surrounded by its receptive field Ω_i and the noise correlation region r_i such that $i \subseteq r_i \subset \Omega_i$. In the following we use subscripts for taking subsets of pixels restricted to a certain domain, so x_{Ω_i} is the part of x that falls in the domain Ω_i . If we train the network f to predict the value x_i given the region x_{Ω_i} as input by minimizing the loss ℓ_i at i then the network will learn to approximate the function $f(x_{\Omega_i}) = \mathbb{E}[x_i \mid x_{\Omega_i}] = x_i$, failing

⁴equal to the covariance of the model prediction and the noise if $f(x)$ and η are both zero-centered

⁵It's still in theory possible to learn the noise by overfitting on very small data. Overfitting happens when we train a large network on small data for a long time, and allows the network to *memorize* the noise (and signal) without ever seeing it as input.

to remove the noise or do anything useful. Now let us remove the region r_i from the input, instead giving the network only the “visible” region $v_i = \Omega_i \setminus r_i$. Training this model approximates the expected value

$$\mathbb{E}[x_i \mid x_{v_i}] = \mathbb{E}[y_i + \eta_i \mid x_{\Omega_i \setminus r_i}]$$

and because of our independence assumptions we can factor this into

$$\mathbb{E}[y_i + \eta_i \mid x_{\Omega_i \setminus r_i}] = \mathbb{E}[y_i \mid x_{\Omega_i \setminus r_i}] + \mathbb{E}[\eta_i \mid x_{\Omega_i \setminus r_i}] \quad .$$

If we assume that $\mathbb{E}[\eta_i \mid x_{\Omega_i \setminus r_i}]$ is zero everywhere⁶ then we see that our training procedure produces the desired expected value,

$$f(x_{v_i}) = \mathbb{E}[x_i \mid x_{v_i}] \approx \mathbb{E}[y_i \mid x] \quad .$$

While in general we might expect local correlations within the noise $\eta_i \not\perp \eta_j$ if pixels i, j are near, we choose the blind mask r_i such that any conditional dependence within the noise is removed, i.e. $\eta_i \perp \eta_{v_i} \mid x_{v_i}$. Thus, the only information about η_i coming from x_{v_i} must come through y_i , which leads to the expected value of zero.

When the noise η_i is uncorrelated with it’s neighbors, i.e. $\eta_i \perp \eta_j \mid x$ for nearby pixels i, j , then it is sufficient to remove only the single pixel i from the visible receptive field giving $r_i = i$, giving us the standard N2V procedure and assumptions. Note that while Noise2Void and StructN2V create blind spots by randomizing pixel values in x_{r_i} it can also be done using a special network architecture e.g. Lehtinen et al., 2018. Finally, the approximate equality $\mathbb{E}[y_i \mid x_{v_i}] \approx y_i$ is only true if we don’t lose too much information about the signal when we remove x_{r_i} . Thus, there is a tension between making r_i too large, which makes the signal y_i harder to predict, and too small, which makes the noise η_i easier to predict.

USING BLIND-SPOT PIXEL INFORMATION

There remains the question of how to use the information in x_{r_i} to improve our estimation of y_i . Batson and Royer (2019) finds that using a simple linear combination of the prediction and input $\alpha f(x) + \beta x$ is able to improve prediction accuracy where the values α, β are determined by the variance of the noise $\text{var}(\eta)$. In *Probabilistic Noise2Void* Krull et al. (2020) follow up on the original Noise2Void technique using multiple fixed acquisitions to observe the pixelwise likelihood⁷ $\Pr(x_i \mid y_i)$ which they

⁶or a constant that can be incorporated into the signal

⁷a model of the noise, η_i

incorporate into a fully probabilistic model equivalent to $P(x_i \mid x_{v_i}; f)$, and train f via by maximizing the likelihood of the observed x_i . While potentially helpful, it is unclear if these extensions are sufficient for our case.

BLIND-SPOT VIA RANDOMIZATION

In blind-spot networks that require randomization⁸ we need to define some procedure

$$R(x) := \mathbf{1}_{v_i}x + \mathbf{1}_{r_i}x' \quad \text{where} \\ x' \sim P_R(x)$$

to randomize the pixels in x_{r_i} before giving an image to the network during training. Here $\mathbf{1}_{v_i}$ is the indicator function, with value 1 inside the domain v_i and 0 outside. Thus, $R(x)$ has values determined exclusively by $P_R(x)$ within the region r_i . The purpose of this procedure is to provide the independence $R(x)_{r_i} \perp\!\!\!\perp x_{v_i} \mid y$.

There are many possible procedures that give us such independence, e.g. we may draw from the uniform distribution $x'_i \sim [0, 1]$, sample a random neighbour, or interpolate by taking a local average excluding x_{r_i} . In theory, there is a tension involved in our choice of R : we want the random image x' to be as similar to x as possible, *while maintaining this independence*. But in practice the technique is not so sensitive to our choice of R . Noise2Void chooses to replace x_i with the value of a random neighbour⁹, while Batson tried sampling a value uniformly over $[0, 1]$ as well as taking a (deterministic) local average excluding x_{r_i} , but both techniques achieve good performance. Finally, while replacing each pixel with a constant value (e.g. zero) does give us the independence, it likely introduces a new dependence between the output $f(R(x))_i$ and this constant value. To be successful at test time, the network must learn to actively ignore the pixels in x_{r_i} no matter what value they have!

HOW DOES THIS RELATE TO TRAINING IN DEEP IMAGE PRIOR?

While Deep Image Prior (DIP, Ulyanov et al. (2018)) is similar in that it uses a CNN to achieve a denoising effect while also using the *raw*, *noisy* image in the loss, it is different in that DIP must be reapplied for every image we want to denoise. DIP uses pure noise as input, instead

⁸as opposed to architecture-based blind spot approaches like Laine et al., 2019

⁹in the original paper, x_i included itself in that neighborhood, despite the dependence that entails.

of a masked version of the noisy target. They then proceed to optimize a similar loss based solely on a single image until the denoising effect is achieved in the network output. This must be repeated for each image! At no point does the network learn to denoise data other than the current target. The network in DIP is blind to both *content* and *noise*, while blind-spot based denoisers attempt to make the network blind to the *noise alone*.

4.5 EXPERIMENTS

We evaluate our STRUCTN2V approach on two fluorescence microscopy datasets, one (*C. majalis*) acquired in 2D with a camera-based spinning disk microscope, the other (*C. elegans*) in 3D with a laser-scanning confocal microscope.

4.5.1 *c. majalis*

We acquired 2D images of fluorescence labeled membranes of a fixed *C. majalis* (Lily of the Valley) sample. All 100 recorded images (1024×1024 pixels) show the same region of interest and only differ in their noise. The average of these 100 images serves as a ground truth (GT) image¹⁰. We train and compare N2V, STRUCTN2V, and N2GT¹¹ using the same neural network architecture for all methods and varying only the training procedure.

CNN architecture We use a 2D U-Net (Ronneberger et al., 2015) with three levels, each consisting of two 3×3 convolutional layers and separated by 2×2 max pooling (constant up-sampling) layers between levels. We train 16 convolution kernels at level one and double (halve) that number after each pooling (up-sampling).

Training

We normalize the input images by applying an (outlier-robust) percentile-based normalization, such that approx 98% of pixel values are in the range [0, 1]. We train and evaluate all models over the entire dataset, as we typically do not make a distinction between train and test images when applying N2V and STRUCTN2V in the wild¹². Note, however, that the performance of N2GT is therefore higher than can be expected in

¹⁰This is possible because we observe that the noise has a spatially constant mean and is uncorrelated between acquisitions.

¹¹Noise-to-Ground-Truth, i.e. conventional supervised training with the GT images as target.

¹²Although over-fitting with blind spot based denoising approaches is possible in theory we have curiously not observed it with N2V or STRUCTN2V in practice, even when training on single images.

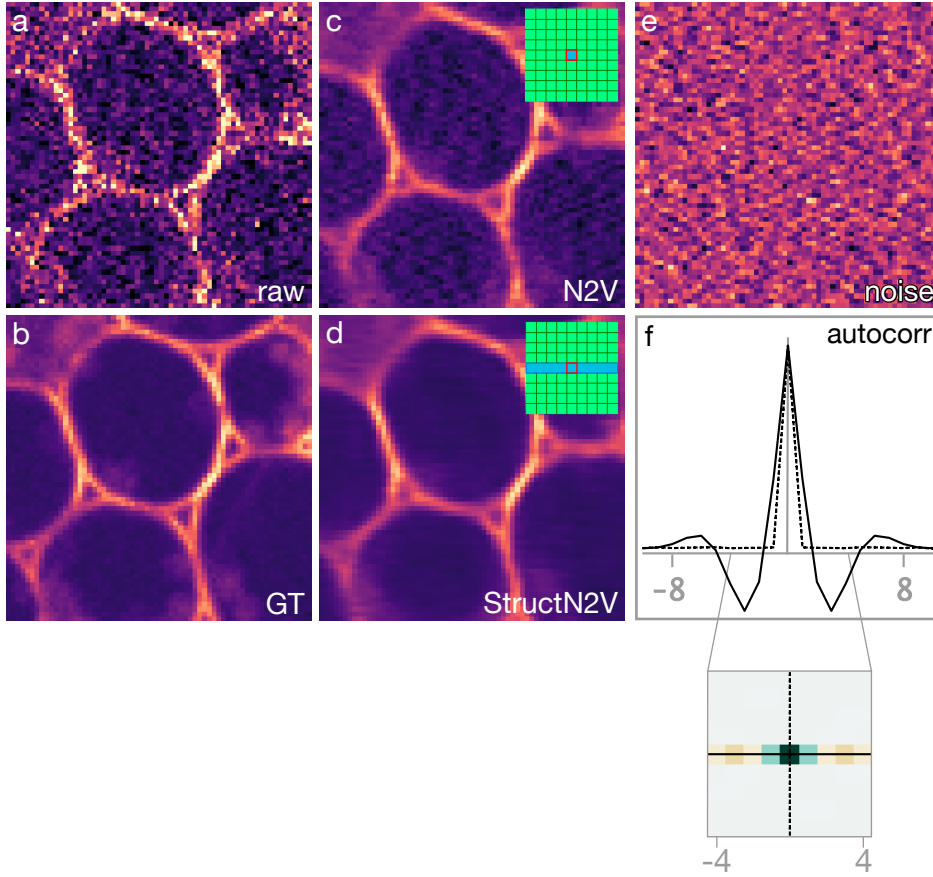


FIGURE 4.6 **STRUCTN2V** removes structured noise from image of fixed *C. majalis* (cell membrane stain/spinning disk). **(a)** Raw **(b)** Ground truth obtained by avg over 100 single frame acquisitions. **(c)** Restoration by N2V. **(d)** Restoration by **STRUCTN2V** (ours). **(e)** Pure noise acquired by darkframe acquisition. Horizontal correlations are barely visible. **(f)** Autocorrelation of pure noise in (e) revealing the horizontal correlations. Insets in (c) and (d) depict the blind masks used during training.

practice, representing an upper bound. We train all methods on 1600 random patches of size 256×256 for 600 epochs (batch size 4) with the Adam optimizer (Kingma and Ba, 2015) at a fixed learning rate of 2×10^{-5} (2×10^{-4} for N2GT).

Masking For N2V and **STRUCTN2V**, we randomly select 2% of pixels as *active* in each training patch. We select mask shapes for **STRUCTN2V** using different methods for each dataset.

(see fig. 4.6f) as described in ???. To that end, we use a horizontal mask of size $w \times 1$, whose width w we vary from small ($w = 3$ pixel) to large ($w = 25$ pixel, see fig. 4.2). In fig. 4.7, we show denoising performance as a function of mask size, and also compare against N2V, N2GT, and standard methods. We find that **STRUCTN2V** provides an improvement over N2V, although some masks are clearly better than others (with the mask 9×1 performing best, see table 4.1). Importantly, the results

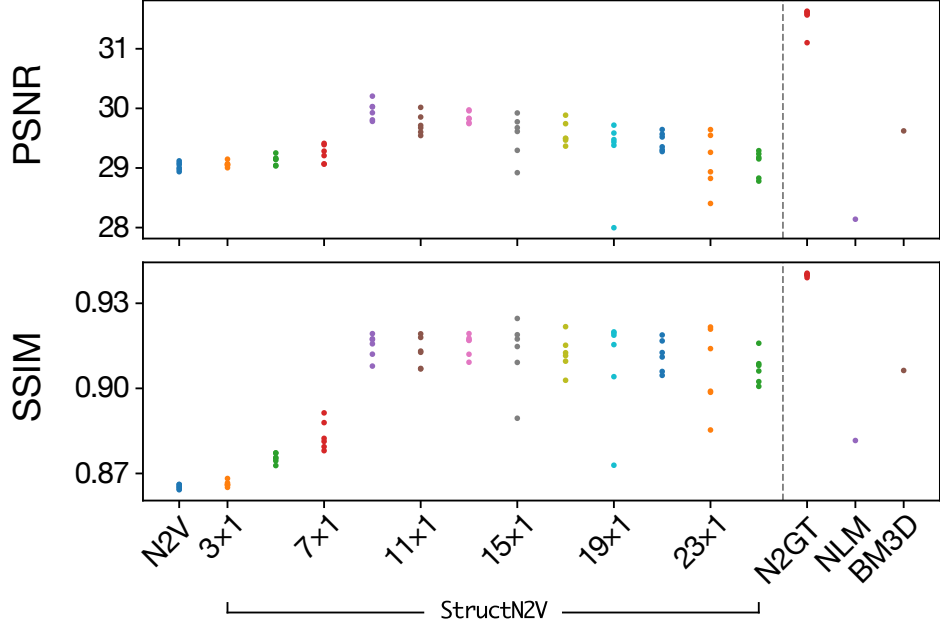


FIGURE 4.7 Comparison of denoising methods on standard image quality metrics PSNR and SSIM (Wang et al., 2004) for *C. majalis* fluorescence images. We use a range of horizontally oriented blind masks of size $w \times 1$ with varying width w , with N2V corresponding to the 1×1 case. Note the wide range of mask sizes with improved performance over the default 1×1 case. For methods with stochastic training (N2V, STRUCTN2V, N2GT) we plot the results from five independent runs. Also, note that each method to the right of the gray line was optimized with knowledge of the ground truth.

of STRUCTN2V show substantially less structured artifacts compared to N2V, as can be inferred from the Fourier power spectrum (fig. 4.3): whereas the spectrum of both the raw (left) and N2V restoration (middle) exhibit vertical stripes at a distinct frequency range (*i.e.* indicating horizontal correlations), they are absent in the STRUCTN2V output (right).

Additionally, table 4.1 summarizes the best results of all methods (with a mask of size 9×1 for STRUCTN2V). Note that NLM, BM3D and N2GT were all tuned/trained to yield optimal results using ground truth data (not available to N2V and STRUCTN2V), hence the results denote upper bounds on what is possible in practice. Still, STRUCTN2V performs better than all compared methods (apart from N2GT, which is expected).

4.5.2 *C. elegans*

We additionally demonstrate that STRUCTN2V can substantially outperform N2V on the publicly available 3D dataset of a developing *C. elegans* (roundworm) embryo from celltrackingchallenge.net (Ulman et al., 2017), which was acquired with a laser-scanning confocal microscope. The data is a time series with 190 frames of size $32 \times 512 \times 708$ (ZYX). The

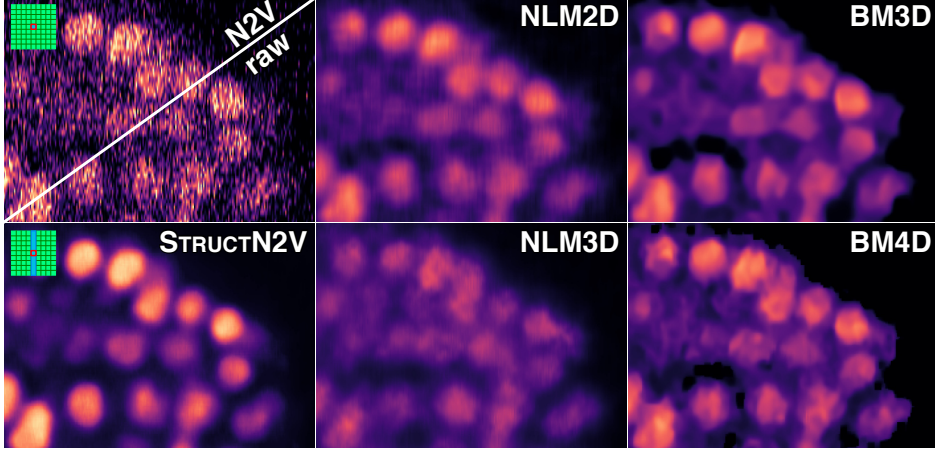


FIGURE 4.8 Qualitative comparison for the 3D *C. elegans* dataset. We show a single slice of the raw stack and the result of NLM (Buades et al., 2005), BM4D (Dabov et al., 2006), N2V (Krull et al., 2019) and STRUCTN2V (ours). The strong vertical correlations in the noise of this image prevent N2V from working, but we a vertically aligned blind mask allow STRUCTN2V to recover a perceptually noise-free image. Insets in N2V and STRUCTN2V depict orientation of the blind masks used for training.

images exhibit visually striking vertical correlations of unknown origin along the Y dimension. In contrast, STRUCTN2V yields smooth and perceptually noise-free images. The mask shape for STRUCTN2V was estimated as $1 \times 17 \times 1$ (ZYX, see fig. 4.2). For all compared methods (N2V and STRUCTN2V) we use a 3D U-Net (Çiçek et al., 2016) with two levels, using two $3 \times 5 \times 5$ (ZYX) convolutional layers followed by $1 \times 2 \times 2$ max pooling per level. Again we learn 16 convolution kernels at the first level and double (halve) that number after each pooling (up-sampling). We also apply the same image normalization as before and train both methods on 1900 random patches of size $32 \times 64 \times 64$ for 100 epochs (batch size 10) using Adam (Kingma and Ba, 2015) at a fixed learning rate of 4×10^{-6} .

Method	PSNR	SSIM
STRUCTN2V (ours)	29.73 ± 0.16	0.913 ± 0.005
N2V (Krull et al., 2019)	29.03 ± 0.06	0.8653 ± 0.0007
NLM (Buades et al., 2005)	28.14	0.8816
BM3D (Dabov et al., 2006)	29.62	0.9063
N2GT	31.52 ± 0.19	0.9399 ± 0.0005

TABLE 4.1 Image quality metrics PSNR and SSIM (Wang et al., 2004) evaluated on *C. majalis* dataset. Note that we include only the best performing STRUCTN2V mask (9×1). For all learned methods (N2V, STRUCTN2V, N2GT) we display the mean ($\pm$ standard deviation) of 5 different training runs using different random seeds.

4.5.3 CA2+ IMAGING IN MOUSE CORTEX

Here we present additional evidence suggesting the `STRUCTN2V` method may provide superior results on a wider variety of acquisition hardware and imaging modalities. We denoise Ca2+ induced fluorescence images of *in vivo* neuronal somata in the mouse cortex acquired via a raster-scanning two-photon microscope. The (horizontal) raster-scanning procedure results in heterogeneous voxel sizes and is likely the cause of the light artifacts revealed by the N2V (see fig. 4.9(e,f)). In fig. 4.9 we compare a variety of denoising methods including N2V, `STRUCTN2V` and some classical heuristics. While all of the blind-spot based approaches provide improved visual quality over classical heuristics, the single-pixel masks of N2V reveal a structured, noisy pattern that is removed by `STRUCTN2V`. We speculate that the revealed pattern is caused by structured noise resulting from the horizontal raster-scan process, as it is removed by horizontal `STRUCTN2V` masks.

Data, Network Architecture and Optimization The training data consists of 100 timepoints of a single 2D section (512x512) of the cortex. Both N2V and `STRUCTN2V` use the same three-level, 2D U-net architecture detailed above in section 4.5.1 for *C. majalis* data trained for 100 epochs with Adam optimizer. Both NLM and bilateral filter methods were optimized by eye using the Napari (Sofroniew et al., 2020) image viewer and the interactive *Magicgui* extension provided by Talley Lambert. Thanks to Horst Obenaus of the Kavli Institute for Systems Neuroscience in Trondheim for providing the data.

4.5.4 NONLINEAR PIXEL RESPONSE

Many sCMOS image sensors exhibit photodetectors with the undesirable property that some small fraction of the pixels don't respond properly to incident light. These malfunctioning photodetectors show up most obviously as individual, bright "hot pixels" in the resulting image. But when we acquire 3D stacks these hot pixels form clearly correlated line segments along the z-axis which are visually distracting and likely harmful for downstream analysis (fig. 4.10 red). While great care is taken by sensor manufacturers to minimize the presence of such malfunctioning photodiodes, they still manage to be a common artifact in bioimages, and computational approaches need to be taken for their removal.

If we could map out the exact locations of the malfunctioning hot pixels on the sensor we could replace the values of these pixels by some function of their neighbors, e.g. via median filtering. However, not all malfunctioning pixels are hot all the time, so in practice it can be difficult

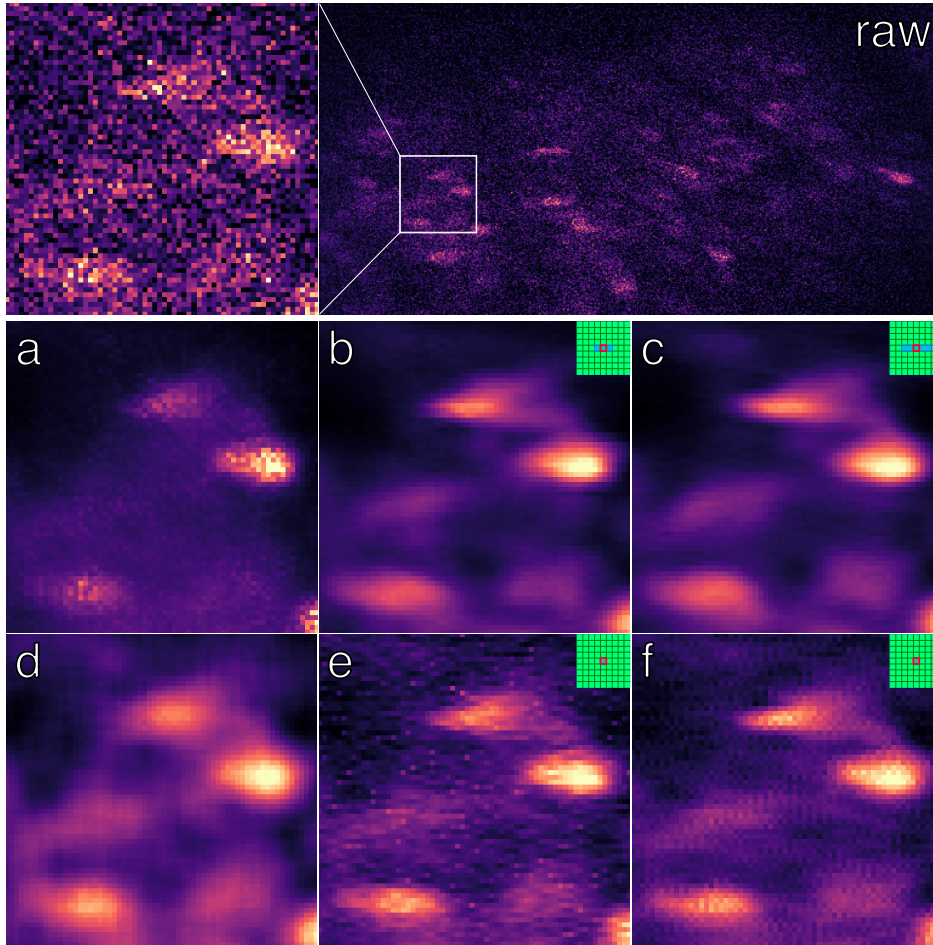


FIGURE 4.9 Comparison of denoising methods on Ca^{2+} induced fluorescence images of mouse cortex. (a) Non-local means (b,c) `STRUCTN2V` with 3×1 and 5×1 mask sizes (d) bilateral filter (e,f) `N2V` with *nearest-neighbour* and *flat distribution* pixel replacement strategies. `N2V` reveals a pattern of structured artifacts removed by `STRUCTN2V`. Bilateral filter (d) removes noise but blurs object boundaries, making object size estimation difficult.

to determine exactly which pixels are misbehaving. It is not uncommon for these artifacts to remain in our images at the time denoising methods are applied, so it is useful to know how their presence influences the denoising process.

In this section we visually explore the performance of three blind-spot based denoisers on fluorescence imaging data with obvious hot pixel artifacts. In theory, there is a problem with `N2V` based methods, as (viewed as a form of noise) hot pixels radically alter the signal and do not satisfy our theoretical criterion $??$. But we may still hope for reasonable performance so long as the hot pixels are sufficiently scarce, as is the case with our dataset.

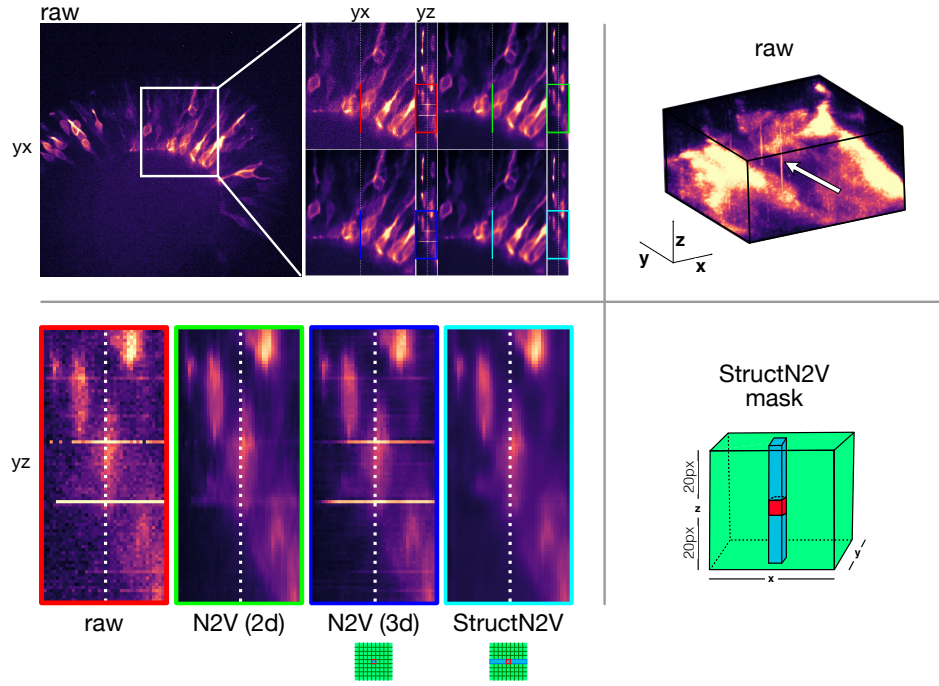


FIGURE 4.10 (left) An sCMOS camera acquisition of fluorescent zebrafish retina. This particular sensor suffers from hot pixels which are visible as bright stripes oriented along the z -axis. 3D N2V (blue) is able to remove the pixelwise independent noise but fails to deal with the hot pixels. 2D N2V (lime) sees these stripes as pixelwise-independent, and is mostly able to remove them. 3D STRUCTN2V (cyan) with a 1px wide z -oriented mask(d) is able to remove the hot stripes almost completely without obvious sacrifices in signal quality using a custom blind mask (bottom right) elongated along the z -axis. (top right) The 3D volume rendering shows z -aligned hot stripes in the raw image.

We show in fig. 4.10 a study of various blind-spot denoisers on fluorescence microscopy data of the developing zebrafish retina. The data has a small percentage of hot pixels which create non-uniform, bright stripes along the z -axis. Each of the N2V-based methods we explore was able to remove the bulk of the pixel-wise independent noise from the data, but interesting differences remain. As in the case of *C. majalis*, normal 3D N2V actually further reveals the underlying visual correlations from the hot pixels, and is completely unable to remove them. We also explore 2D N2V for this data, with training and prediction only on 2D slices of the 3D data. Restricting the network to a purely 2D view of these hot stripes prevents the net from learning the hot pixel z -axis correlations, and largely removes the noise. The faint residual stripes hint at subtle within-plane correlations between neighboring photodetectors on the sCMOS chip. Finally, STRUCTN2V is able to remove the hot stripes almost entirely, along with the pixel-wise independent noise, producing a visually appealing smooth result. By using the mask shown in fig. 4.10d we enable STRUCTN2V to use the full 3D context for predicting the signal, as opposed to just a 2D slice, without making the noise easier to predict.

Training Details Each of the approaches contrasted above was trained on a single 3D stack with shape $40 \times 1024 \times 1040$ (*zyx*), using the same U-net architecture detailed in the *C. elegans* section (*C. majalis* section for 2D net). The kernel used for creating the `STRUCTN2V` mask in this case was 1px wide in *x,y* but extended along the *z*-axis to the entire depth of the receptive field, to prevent any correlations along that line, however distant, from being learned by the network (see fig. 4.10 d).

4.6 DISCUSSION

We proposed `STRUCTN2V`, an extension of self-supervised blind spot networks, which allows us to effectively remove spatially correlated (structured) noise without the need to acquire clean ground truth images for training. We demonstrate the capabilities of this approach on several fluorescence microscopy datasets involving a diversity of specimens, microscope hardware, and various noise correlation patterns. In many cases we are able to show denoising quality which is an improvement over both modern blind-spot based CNN approaches as well as popular heuristic approaches.

We discussed several heuristics to select a good mask shape, which we found effective in practice. Nevertheless, selecting an appropriate mask shape requires nontrivial user knowledge which may complicate the adoption of our method by the wider community. In future work we hope to explore a variety of applications of the method, as well as potential techniques for simplifying/automating mask shape selection.

DISCUSSION

In this thesis we introduced three new models to help deal with common problems in bioimage informatics.

In Chapter 2 we introduced CP-Net , a CNN for cell and nucleus detection that requires only centerpoint dot annotations for training.

First, we showed that this model is competitive with or exceeds current state of the art detection approaches on many of the ISBI Cell Segmentation Challenge datasets. However, CP-Net is still data hungry, with some datasets requiring tens of thousands of annotations before performance saturates (see fig. 2.22). This number should reflect both the inherent diversity (difficulty) in a given training set as well as the learning ability of our network. A more careful study would involve retraining multiple times on different random subsamples of each dataset to study how scores respond as a function of training data size while keeping the dataset constant. Perhaps with more advanced data augmentation we could improve performance, as the models discussed in this thesis only used the standard set of rotations and flips. While some additional experiments trained on data augmented with elastic deformations, artificial noise, and brightness and contrast we found that these steps had a negligible effect on performance.

Second, while CP-Net is capable of detecting elongated, non-circular objects an undertrained network is prone to false positives and false negatives at a higher rate compared to circular or spherical objects. This makes intuitive sense as the receptive field of CP-Net is round, and so accomodates round objects most easily. CP-Net will always struggle with filamentous objects much larger than the receptive field without any clear visual marker associated with their center, e.g. a neuron without any clear cell body. Further work could explore the interplay of receptive shape, kernel shape, and non maximum suppression to better adapt to oblong cell shapes.

Third, there is the fundamental ambiguity of some cases, as demonstrated by high inter-annotator variability on some ISBI datasets. In future work it may be preferable to mark ambiguous objects explicitly

with a special label acknowledging them as such. These ambiguous cases could then be optionally be incorporated into model training and evaluation. A model could ignore such cases or attempt to predict a confidence level for each object which should be low on ambiguous cases. These could then be left out of downstream model performance evaluation or included with a unique weighting.

In Chapter 3 we extended CP-Net with a parameter-free object linking model to perform cell tracking and evaluated it by participating in the ISBI Cell Tracking Challenge. We found that a very simple linking model can perform well if it is given ground truth cell centerpoint detections as input. This inspired us to use CP-Net, an accurate cell centerpoint prediction model, in combination with the “nearest parent” linker to tracking cell. We found this combination outperforms many models with more complex linking steps through the sheer quality of the initial detections, and achieved first place scores in the ISBI Cell Tracking Challenge on three datasets of developing embryos.

This nearest neighbour linking model is intentionally naive, and could almost certainly buy accuracy at the cost of additional parameters characterizing (potentially learned) priors on the distribution of cell displacements, the frequency of divisions, the apparent cell sizes, etc. However, we believe the speed and simplicity of the model has inherent value in that it requires no manual or automated parameter tuning, is easy to understand and implement by 3rd parties, and fits easily into interactive applications where feedback must be on the timescale of human attention.

In Chapter 3 we also extend the Tracking By Assignment framework for cell tracking with “velocity gradient costs” to aid in tracking when cell motions are locally correlated and displacements are large relative to the cell nearest neighbour spacing. We were able to show that through the addition of these costs we could improve the performance of the tracker on two real datasets: *Schmidtea mediterranea* and human breast cancer cells. Introducing the costs requires choosing a correlation decay lengthscale and cutoff length parameters, but they are intuitive, and in our experiments we were able to find good results without automated optimization.

However, this approach introduces a large number of additional variables to the ILP formulation of the tracking problem, which caused significant slowdown in Gurobi’s time to find an optimal solution. While it might be possible to tune Gurobi’s settings in a way that does a better job balancing solution accuracy and solver time we couldn’t find any acceptable parameters that didn’t involve a large slowdown. If this model proves valuable it may be worth designing a model-specific solver or a

fast heuristic solver. Still, this tradeoff of buying accuracy in exchange for compute time is often acceptable and the gains in accuracy were significant in the datasets we explored.

In Chapter 4 we introduced the `STRUCTN2V` denoising technique for self-supervised denoising of images with locally-correlated *structured* noise. As a straightforward extension of *Noise2Void*, this approach is independent of the specific CNN architecture. We showed that it provides a significant visual and numerical improvement on multiple, real fluorescence microscopy datasets in 2D and 3D. Further, we showed how a 3D blind mask extended along the Z dimension can be used to remove hot pixels from 3D data.

Although the training mask is potentially a complex hyperparameter we found it straightforward to determine in practice for the cases studied here. Unlike some other denoising models including *Noise2Void* variants `STRUCTN2V` does not incorporate an explicit noise model. Noise models tend to improve performance, allow for MAP estimation of the signal, and sampling from the posterior, so including one into `STRUCTN2V` is potentially interesting future work.

Part I

APPENDIX

APPENDIX

A

Name	radius	zoom	sigmas	nms_shape	train_patch_shape	isbi_border
BF-C2DL-HSC	10	[1, 1]	[5, 5]	[5, 5]	[256 256]	[25, 25]
BF-C2DL-MuSC	10	[1, 1]	[5, 5]	[5, 5]	[256 256]	[25, 25]
DIC-C2DH-HeLa	20	[0.5, 0.5]	[5, 5]	[5, 5]	[256 256]	[50, 50]
Fluo-C2DL-MSC	40	[0.25, 0.25]	[5, 5]	[5, 5]	[256 256]	[50, 50]
Fluo-N2DH-GOWT1	20	[0.5, 0.5]	[5, 5]	[5, 5]	[256 256]	[50, 50]
Fluo-N2DH-SIM+	20	[1, 1]	[5, 5]	[5, 5]	[256 256]	[0, 0]
Fluo-N2DL-HeLa	10	[1, 1]	[5, 5]	[5, 5]	[256 256]	[25, 25]
PhC-C2DH-U373	20	[0.5, 0.5]	[5, 5]	[5, 5]	[256 256]	[50, 50]
PhC-C2DL-PSC	7	[1, 1]	[3, 3]	[3, 3]	[256 256]	[25, 25]
Fluo-C3DH-A549	20	[1, 0.5, 0.5]	[3, 5, 5]	[3, 5, 5]	[16 128 128]	[0, 0, 0]
Fluo-C3DH-A549-SIM	20	[1, 0.5, 0.5]	[3, 5, 5]	[3, 5, 5]	[16 128 128]	[0, 0, 0]
Fluo-C3DH-H157	80	[0.25, 0.125, 0.125]	[1, 3, 3]	[3, 5, 5]	[16 128 128]	[0, 50, 50]
Fluo-C3DL-MDA231	10	[1, 1, 1]	[3, 5, 5]	[3, 5, 5]	[16 128 128]	[0, 25, 25]
Fluo-N3DH-CE	20	[1, 0.5, 0.5]	[3, 5, 5]	[3, 5, 5]	[16 128 128]	[0, 50, 50]
Fluo-N3DH-CHO	40	[1, 0.25, 0.25]	[3, 5, 5]	[3, 5, 5]	[16 128 128]	[0, 50, 50]
Fluo-N3DH-SIM+	20	[1, 0.5, 0.5]	[3, 5, 5]	[3, 5, 5]	[16 128 128]	[0, 0, 0]
Fluo-N3DL-DRO	7	[1, 1, 1]	[1.5, 3, 3]	[3, 5, 5]	[16 128 128]	[0, 0, 0]
Fluo-N3DL-TRIC	10	[1, 1, 1]	[3, 5, 5]	[3, 5, 5]	[16 128 128]	[0, 0, 0]
Fluo-N3DL-TRIF	10	[1, 1, 1]	[3, 3, 3]	[3, 5, 5]	[64 64 64]	[0, 0, 0]

TABLE A.1 Table of CP-Net parameters for training and inference on datasets for the ISBI Cell Tracking Challenge and ISBI Cell Segmentation Benchmark.

A.1 ISBI CELL TRACKING CHALLENGE DATASET DESCRIPTIONS

For more information see celltrackingchallenge.net 2D datasets and 3D datasets.

BF-C2DL-HSC

Mouse hematopoietic stem cells in hydrogel microwells

Dr. H. Blau, Baxter Laboratory for Stem Cell Biology, Stanford University, USA

Microscope: Zeiss PALM/AxioObserver Z1

Objective lens: EC Plan-Neofluar 10x/0.30 Ph1

Pixel size (microns): 0.645 x 0.645

Time step (min): 5

BF-C2DL-MuSC

Mouse muscle stem cells in hydrogel microwells

Dr. H. Blau, Baxter Laboratory for Stem Cell Biology, Stanford University, USA

Microscope: Zeiss PALM/AxioObserver Z1

Objective lens: EC Plan-Neofluar 10x/0.30 Ph1

Pixel size (microns): 0.645 x 0.645

Time step (min): 5

DIC-C2DH-HeLa

HeLa cells on a flat glass

Dr. G. van Cappellen. Erasmus Medical Center, Rotterdam, The Netherlands

Microscope: Zeiss LSM 510 Meta

Objective lens: Plan-Apochromat 63x/1.4 (oil)

Pixel size (microns): 0.19 x 0.19

Time step (min): 10

Fluo-C2DL-Huh7

Human hepatocarcinoma-derived cells expressing the fusion protein YFP-TIA-1

Microscope: Nikon Eclipse Ti2

Objective lens: CFI Plan Apo Lambda 20x/0.75

Pixel size (microns): 0.65 x 0.65

Time step (min): 15

Fluo-C2DL-MSC

Rat mesenchymal stem cells on a flat polyacrylamide substrate

Dr. F. Prósper. Cell Therapy laboratory, Center for Applied Medical Research (CIMA), Pamplona, Spain

Microscope: PerkinElmer UltraVIEW ERS

Objective lens: Plan-Neofluar 10x/0.3 (Plan-Apo 20x/0.75)

Pixel size (microns): 0.3 x 0.3 (0.3977 x 0.3977)

Time step (min): 20 (30)

Fluo-N2DH-GOWT1

GFP-GOWT1 mouse stem cells

Dr. E. Bártořá. Institute of Biophysics, Academy of Sciences of the Czech Republic, Brno, Czech Republic

Microscope: Leica TCS SP5

Objective lens: Plan-Apochromat 63x/1.4 (oil)

Pixel size (microns): 0.240 x 0.240

Time step (min): 5

Fluo-N2DL-HeLa

HeLa cells stably expressing H2b-GFP

Mitochcek Consortium Microscope: Olympus IX81

Objective lens: Plan 10x/0.4

Pixel size (microns): 0.645 x 0.645

Time step (min): 30

PhC-C2DH-U373

Glioblastoma-astrocytoma U373 cells on a polyacrylamide substrate

Dr. S. Kumar. Department of Bioengineering, University of California at Berkeley, Berkeley CA (USA)

Microscope: Nikon

Objective lens: Plan Fluor DLL 20x/0.5

Pixel size (microns): 0.65 x 0.65

Time step (min): 15

PhC-C2DL-PSC

Pancreatic stem cells on a polystyrene substrate

Dr. T. Becker and Dr. D. Rapoport. Fraunhofer Institution for Marine Biotechnology, Lübeck, Germany

Microscope: Olympus ix-81

Objective lens: UPLFLN 4XPH

Pixel size (microns): 1.6 x 1.6

Time step (min): 10

Fluo-N2DH-SIM+

Simulated nuclei of HL60 cells stained with Hoescht

Dr. V. Uľman and Dr. D. Svoboda. Centre for Biomedical Image Analysis (CBIA),

Microscope: Zeiss Axiovert 100S with a Micromax 1300-YHS camera

Objective lens: Plan-Apochromat 40x/1.3 (oil)

Pixel size (microns): 0.125 x 0.125

Time step (min): 29

Fluo-C3DH-A549

GFP-actin-stained A549 Lung Cancer cells embedded in a Matrigel matrix

Dr. A. Rouzaut. Cell Adhesion and Metastasis Laboratory, Center for Applied Medical Research (CIMA), Pamplona, Spain

Microscope: PerkinElmer UltraVIEW ERS

Objective lens: Plan-Apochromat 63x/1.2 (water)

Voxel size (microns): 0.126 x 0.126 x 1.0

Time step (min): 2

Fluo-C3DH-H157

GFP-transfected H157 Lung Cancer cells embedded in a matrigel matrix

Dr. A. Rouzaut. Cell Adhesion and Metastasis Laboratory, Center for Applied Medical Research (CIMA), Pamplona, Spain

Microscope: PerkinElmer UltraVIEW ERS

Objective lens: Plan-Apochromat 63x/1.2 (water)

Voxel size (microns): 0.126 x 0.126 x 0.5

Time step (min): 2 (1)

Fluo-C3DL-MDA231

MDA231 human breast carcinoma cells infected with a pMSCV vector including the GFP sequence, embedded in a collagen matrix

Dr. R. Kamm. Dept. of Biological Engineering, Massachusetts Institute of Technology, Cambridge MA (USA)

Microscope: Olympus FluoView F1000

Objective lens: Plan 20x/0.7

Voxel size (microns): 1.242 x 1.242 x 6.0

Time step (min): 80

Fluo-N3DH-CE

C.elegans developing embryo

Waterston Lab, University of Washington, Seattle, WA, USA

Microscope: Zeiss LSM 510 Meta

Objective lens: Plan-Apochromat 63x/1.4 (oil)

Voxel size (microns): 0.09 x 0.09 x 1.0

Time step (min): 1 (1.5)

Fluo-N3DH-CHO

Chinese Hamster Ovarian (CHO) nuclei overexpressing GFP-PCNA

Dr. J. Essers. Dept. of Cell Biology, Erasmus Medical Center, Rotterdam, The Netherlands

Microscope: Zeiss LSM 510

Objective lens: Plan-Apochromat 63x/1.4 (oil)

Voxel size (microns): 0.202 x 0.202 x 1.0

Time step (min): 9.5

Fluo-N₃DL-DRO

Developing *Drosophila Melanogaster* embryo

Dr. P. Keller. Howard Hughes Medical Institute, Janelia Farms Research Campus, Ashburn VA, USA

Microscope: SIMView light-sheet microscope

Objective lens: 16x/0.8 (water)

Voxel size (microns): 0.406 x 0.406 x 2.03

Time step (sec): 30

Fluo-N₃DL-TRIC

Developing *Tribolium Castaneum* embryo (3D cartographic projection)

Dr. A. Jain. Max Planck Institute of Molecular Cell Biology and Genetics, Dresden, Germany

Microscope: Zeiss LightSheet LZ.1

Objective lens: Plan-Apochromat 20x/1.0 (water)

Voxel size (microns): NA due to cartographic projections

Time step (min): 1.5

Fluo-N₃DL-TRIF

Developing *Tribolium Castaneum* embryo

Dr. A. Jain. Max Planck Institute of Molecular Cell Biology and Genetics, Dresden, Germany

Microscope: Zeiss LightSheet LZ.1

Objective lens: Plan-Apochromat 20x/1.0 (water)

Voxel size (microns): 0.38 x 0.38 x 0.38

Time step (min): 1.5

Fluo-C₃DH-A549-SIM

Simulated GFP-actin-stained A549 Lung Cancer cells embedded in a Matrigel matrix

Dr. M. Maška and Dr. D. V. Sorokin. Centre for Biomedical Image Analysis (CBIA),

Microscope: PerkinElmer UltraVIEW ERS

Objective lens: Plan-Apochromat 63x/1.2 (water)

Voxel size (microns): 0.126 x 0.126 x 1.0

Time step (sec): 20

Fluo-N₃DH-SIM+

Simulated nuclei of HL60 cells stained with Hoechst

Dr. V. Ulman and Dr. D. Svoboda. Centre for Biomedical Image Analysis (CBIA),

Microscope: Zeiss Axiovert 100S with a Micromax 1300-YHS camera

Objective lens: Plan-Apochromat 40x/1.3 (oil)

Voxel size (microns): 0.125 x 0.125 x 0.200

Time step (min): 29

ERKLÄRUNGEN ZUR ERÖFFNUNG DES PROMOTIONSVERFAHRENS

1. Hiermit versichere ich, dass ich die vorliegende Arbeit ohne unzulässige Hilfe Dritter und ohne Benutzung anderer als der angegebenen Hilfsmittel angefertigt habe; die aus fremden Quellen direkt oder indirekt übernommenen Gedanken sind als solche kenntlich gemacht.
2. Bei der Auswahl und Auswertung des Materials sowie bei der Herstellung des Manuskripts habe ich Unterstützungsleistungen von folgenden Personen erhalten:
3. Weitere Personen waren an der geistigen Herstellung der vorliegenden Arbeit nicht beteiligt. Insbesondere habe ich nicht die Hilfe eines kommerziellen Promotionsberaters in Anspruch genommen. Dritte haben von mir weder unmittelbar noch mittelbar geldwerte Leistungen für Arbeiten erhalten, die im Zusammenhang mit dem Inhalt der vorgelegten Dissertation stehen.
4. Die Arbeit wurde bisher weder im Inland noch im Ausland in gleicher oder ähnlicher Form einer anderen Prüfungsbehörde vorgelegt und ist auch noch nicht veröffentlicht worden.
5. Ich bestätige, dass ich die geltende Promotionsordnung der Fakultät Informatik der Technischen Universität Dresden anerkenne.

ACKNOWLEDGEMENTS

The work on display in this thesis is a product of the efforts of many kind and brilliant people. First, I would like to thank my supervisor *Gene Myers* for the opportunity to do this work and for convincing me that tracking cells during embryogenesis was a fascinating problem. Thank you to *Ivo Sbalzarini* and *Pavel Tomacak* for your guidance and criticism as a part of my thesis advisory committee.

Thank you to my collaborators for your beautiful images and for reminding me why bioimage informatics matters: *Maurício Rocha-Martins*, *Tobias Boothe*, *Horst Obenhaus*, *Carine Stapel*, and *Daniela Vorkel*.

Thank you to *ISBI Cell Tracking Challenge team* and in particular *Vladimír Ulman* and *Martin Maška* for organizing a tremendously useful competition, making an incredible diversity of datasets public, answering my questions, and guiding me through the submission process.

Thank you to the CBG computer department and especially *Oscar Gonzalez* for managing the computer cluster and teaching me how to use it.

Thank you to my mentors for educating me and forcing me to get things done, sometimes, *Martin Weigert*, *Uwe Schmidt*, *Alex Krull*, and *Florian Jug*.

Thank you again, Martin, for helping me see this thesis through, at long last, to its completion.

Thank you to my colleagues for thoughtful discussions and good humor. You made work a joy every day. *Laurent Abouchar*, *Alex Dibrov*, *Suhrid Ghosh*, *Debayan Saha*, *Fabio Cunial*, *Kira Vinogradova*, *The Jug Lab*, *Corinna Blasse*, *Arne Ludwig*, *Robert Haase*, *Loïc Royer*, *Dagmar Kainmueller*, *Nicola Maghelli*, *David Chen*, *German Tischler*, *Martin Pippel*, and *Dave Richmond*.

Thank you to *Reni Schimmel* who took care of the Myers lab with a smile. Thank you to my friends inside and outside of the CBG who made life in Dresden more fun than should be allowed, really. Finally, thank you to my family for your love and unwavering support: Mom and Dad, Callie, Finley, Suzanne, Tony, Max and Luke.

In memory of Suzanne, who introduced me to cell tracking in 2010 and is a part of this work in so many ways.

BIBLIOGRAPHY

- Abramoff, M. D., P. J. Magalhães, and S. J. Ram (2004). “Image Processing with ImageJ”. In: *Biophotonics international* 11.7, pp. 36–42 (cit. on p. 2).
- Aigouy, B., R. Farhadifar, D. B. Staple, A. Sagner, J.-C. Röper, F. Jülicher, and S. Eaton (2010). “Cell Flow Reorients the Axis of Planar Polarity in the Wing Epithelium of *Drosophila*”. In: *Cell* 142.5, pp. 773–786. DOI: 10.1016/j.cell.2010.07.042 (cit. on p. 41).
- Akram, S. U., J. Kannala, L. Eklund, and J. Heikkilä (2016a). “Cell Proposal Network for Microscopy Image Analysis”. In: *2016 IEEE International Conference on Image Processing (ICIP)*. 2016 IEEE International Conference on Image Processing (ICIP), pp. 3199–3203. DOI: 10.1109/ICIP.2016.7532950 (cit. on p. 9).
- (2016b). “Joint Cell Segmentation and Tracking Using Cell Proposals”. In: *2016 IEEE 13th International Symposium on Biomedical Imaging (ISBI)*. 2016 IEEE 13th International Symposium on Biomedical Imaging (ISBI), pp. 920–924. DOI: 10.1109/ISBI.2016.7493415 (cit. on p. 9).
- (2017). “Cell Tracking via Proposal Generation and Selection”. arXiv: 1705.03386 (cit. on p. 9).
- Amat, F. and P. J. Keller (2013). “Towards Comprehensive Cell Lineage Reconstructions in Complex Organisms Using Light-Sheet Microscopy”. In: *Development, Growth & Differentiation* 55.4, pp. 563–578. DOI: 10.1111/dgd.12063 (cit. on p. 42).
- Amat, F., W. Lemon, D. P. Mossing, K. McDole, Y. Wan, K. Branson, E. W. Myers, and P. J. Keller (2014). “Fast, Accurate Reconstruction of Cell Lineages from Large-Scale Fluorescence Microscopy Data”. In: *Nature Methods* 11.9 (9), pp. 951–958. DOI: 10.1038/nmeth.3036 (cit. on pp. 13, 27, 28, 42, 44, 45, 64).
- Amat, F., E. W. Myers, and P. J. Keller (2013). “Fast and Robust Optical Flow for Time-Lapse Microscopy Using Super-Voxels”. In: *Bioinformatics* 29.3, pp. 373–380. DOI: 10.1093/bioinformatics/bts706 (cit. on p. 55).
- Arteta, C., V. Lempitsky, J. A. Noble, and A. Zisserman (2012). “Learning to Detect Cells Using Non-overlapping Extremal Regions”. In: *Medical Image Computing and Computer-Assisted Intervention – MICCAI 2012*. Ed. by N. Ayache, H. Delingette, P. Golland, and K. Mori. Lecture Notes in Computer Science. Berlin, Heidelberg: Springer, pp. 348–356. DOI: 10.1007/978-3-642-33415-3_43 (cit. on p. 9).
- Arzt, M., J. Deschamps, C. Schmied, T. Pietzsch, D. Schmidt, P. Tomancak, R. Haase, and F. Jug (2022). “LABKIT: Labeling and Segmentation Toolkit for Big Image Data”. In: *Frontiers in Computer Science* 4 (cit. on p. 13).

- Ballard, D. H. (1981). "Generalizing the Hough Transform to Detect Arbitrary Shapes". In: *Pattern Recognition* 13.2, pp. 111–122. DOI: 10.1016/0031-3203(81)90009-1 (cit. on p. 13).
- Bao, Z., J. I. Murray, T. Boyle, S. L. Ooi, M. J. Sandel, and R. H. Waterston (2006). "Automated Cell Lineage Tracing in *Caenorhabditis Elegans*". In: *Proceedings of the National Academy of Sciences* 103.8, pp. 2707–2712. DOI: 10.1073/pnas.0511111103. pmid: 16477039 (cit. on pp. 13, 27, 28, 42, 45).
- Batchen, G., M. Daniel, and S. Pinson (2004). "Light and Dark: The Daguerreotype and Art History". In: *The Art Bulletin* 86.4, p. 764. DOI: 10.2307/4134463. JSTOR: 4134463 (cit. on pp. 1, 2).
- Batson, J. and L. Royer (2019). "Noise2Self: Blind Denoising by Self-Supervision". In: *International Conference on Machine Learning*. International Conference on Machine Learning, pp. 524–533 (cit. on pp. 65, 72).
- Berg, S., D. Kutra, T. Kroeger, C. N. Straehle, B. X. Kausler, C. Haubold, M. Schiegg, J. Ales, T. Beier, M. Rudy, K. Eren, J. I. Cervantes, B. Xu, F. Beuttenmueller, A. Wolny, C. Zhang, U. Koethe, F. A. Hamprecht, and A. Kreshuk (2019). "Ilastik: Interactive Machine Learning for (Bio)Image Analysis". In: *Nature Methods* 16.12 (12), pp. 1226–1232. DOI: 10.1038/s41592-019-0582-9 (cit. on pp. 11, 54).
- Betzig, E., G. H. Patterson, R. Sougrat, O. W. Lindwasser, S. Olenych, J. S. Bonifacino, M. W. Davidson, J. Lippincott-Schwartz, and H. F. Hess (2006). "Imaging Intracellular Fluorescent Proteins at Nanometer Resolution". In: *Science* 313.5793, pp. 1642–1645. DOI: 10.1126/science.1127344. pmid: 16902090 (cit. on pp. 1, 14).
- Beucher, S. and C. Lantuéjoul (1979). "Use of Watersheds in Contour Detection". In: *Proceedings of the International Workshop on Image Processing, Real Time Edge and Motion Detection/Estimation*. Rennes, France (cit. on p. 14).
- Bise, R., Z. Yin, and T. Kanade (2011). "Reliable Cell Tracking by Global Data Association". In: *2011 IEEE International Symposium on Biomedical Imaging: From Nano to Macro*. 2011 IEEE International Symposium on Biomedical Imaging: From Nano to Macro, pp. 1004–1010. DOI: 10.1109/ISBI.2011.5872571 (cit. on p. 45).
- Blasse, C. (2016). "TOWARDS ACCURATE AND EFFICIENT CELL TRACKING DURING FLY WING DEVELOPMENT". Technische Universitaet Dresden. 144 pp. (cit. on pp. 12, 55).
- Boser, B. E., I. M. Guyon, and V. N. Vapnik (1992). "A Training Algorithm for Optimal Margin Classifiers". In: *Proceedings of the Fifth Annual Workshop on Computational Learning Theory*. COLT '92. New York, NY, USA: Association for Computing Machinery, pp. 144–152. DOI: 10.1145/130385.130401 (cit. on p. 9).
- Boyle, T. J., Z. Bao, J. I. Murray, C. L. Araya, and R. H. Waterston (2006). "AceTree: A Tool for Visual Analysis of *Caenorhabditis Elegans* Embryogenesis". In: *BMC Bioinformatics* 7.1, p. 275. DOI: 10.1186/1471-2105-7-275 (cit. on p. 13).
- Boyle, W. S. and G. E. Smith (1970). "Charge Coupled Semiconductor Devices". In: *Bell System Technical Journal* 49.4, pp. 587–593. DOI: 10.1002/j.1538-7305.1970.tb01790.x (cit. on p. 1).

- Breiman, L. (2001). "Random Forests". In: *Machine learning* 45.1, pp. 5–32 (cit. on p. 13).
- Buades, A., B. Coll, and J.-M. Morel (2005). "A Non-Local Algorithm for Image Denoising". In: *2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05)*. 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05). Vol. 2. San Diego, CA, USA: IEEE, pp. 60–65. DOI: 10.1109/CVPR.2005.38 (cit. on pp. 64, 77).
- Cahill, N. D., J. A. Noble, and D. J. Hawkes (2009a). "A Demons Algorithm for Image Registration with Locally Adaptive Regularization". In: *Medical Image Computing and Computer-Assisted Intervention – MICCAI 2009*. Ed. by G.-Z. Yang, D. Hawkes, D. Rueckert, A. Noble, and C. Taylor. Lecture Notes in Computer Science. Berlin, Heidelberg: Springer, pp. 574–581. DOI: 10.1007/978-3-642-04268-3_71 (cit. on p. 55).
- (2009b). "Demons Algorithms for Fluid and Curvature Registration". In: *2009 IEEE International Symposium on Biomedical Imaging: From Nano to Macro*. 2009 IEEE International Symposium on Biomedical Imaging: From Nano to Macro, pp. 730–733. DOI: 10.1109/ISBI.2009.5193151 (cit. on p. 55).
- Carion, N., F. Massa, G. Synnaeve, N. Usunier, A. Kirillov, and S. Zagoruyko (2020). "End-to-End Object Detection with Transformers". In: *Computer Vision – ECCV 2020*. Ed. by A. Vedaldi, H. Bischof, T. Brox, and J.-M. Frahm. Lecture Notes in Computer Science. Cham: Springer International Publishing, pp. 213–229. DOI: 10.1007/978-3-030-58452-8_13 (cit. on p. 11).
- Çiçek, Ö., A. Abdulkadir, S. S. Lienkamp, T. Brox, and O. Ronneberger (2016). "3D U-Net: Learning Dense Volumetric Segmentation from Sparse Annotation". In: *Medical Image Computing and Computer-Assisted Intervention – MICCAI 2016*. Ed. by S. Ourselin, L. Joskowicz, M. R. Sabuncu, G. Unal, and W. Wells. Lecture Notes in Computer Science. Cham: Springer International Publishing, pp. 424–432. DOI: 10.1007/978-3-319-46723-8_49 (cit. on pp. 16, 77).
- Ciresan, D., A. Giusti, L. Gambardella, and J. Schmidhuber (2012). "Deep Neural Networks Segment Neuronal Membranes in Electron Microscopy Images". In: *Advances in Neural Information Processing Systems* 25 (cit. on pp. 4, 9).
- Cireşan, D. C., A. Giusti, L. M. Gambardella, and J. Schmidhuber (2013). "Mitosis Detection in Breast Cancer Histology Images with Deep Neural Networks". In: *Medical Image Computing and Computer-Assisted Intervention – MICCAI 2013*. Ed. by K. Mori, I. Sakuma, Y. Sato, C. Barillot, and N. Navab. Lecture Notes in Computer Science. Berlin, Heidelberg: Springer, pp. 411–418. DOI: 10.1007/978-3-642-40763-5_51 (cit. on pp. 4, 9, 10).
- Cplex, I. I. (2009). "V12. 1: User's Manual for CPLEX". In: *International Business Machines Corporation* 46.53, p. 157 (cit. on p. 54).
- Dabov, K., A. Foi, V. Katkovnik, and K. Egiazarian (2006). "Image Denoising with Block-Matching and 3D Filtering". In: *Image Processing: Algorithms and Systems, Neural Networks, and Machine Learning*. Image Processing: Algorithms and Systems, Neural Networks, and Machine Learning. Vol. 6064. International Society for Optics and Photonics, p. 606414. DOI: 10.1117/12.643267 (cit. on pp. 64, 77).

- De Chaumont, F., S. Dallongeville, N. Chenouard, N. Hervé, S. Pop, T. Provoost, V. Meas-Yedid, P. Pankajakshan, T. Lecomte, Y. Le Montagner, T. Lagache, A. Dufour, and J.-C. Olivo-Marin (2012). "Icy: An Open Bioimage Informatics Platform for Extended Reproducible Research". In: *Nature Methods* 9.7 (7), pp. 690–696. DOI: 10.1038/nmeth.2075 (cit. on p. 2).
- Doersch, C., A. Gupta, and A. A. Efros (2015). "Unsupervised Visual Representation Learning by Context Prediction". In: Proceedings of the IEEE International Conference on Computer Vision, pp. 1422–1430 (cit. on p. 65).
- Dzyubachyk, O., W. A. van Cappellen, J. Essers, W. J. Niessen, and E. Meijering (2010). "Advanced Level-Set-Based Cell Tracking in Time-Lapse Fluorescence Microscopy". In: *IEEE Transactions on Medical Imaging* 29.3, pp. 852–867. DOI: 10.1109/TMI.2009.2038693 (cit. on p. 45).
- Eils, R. and C. Athale (2003). "Computational Imaging in Cell Biology". In: *Journal of Cell Biology* 161.3, pp. 477–481. DOI: 10.1083/jcb.200302097 (cit. on p. 41).
- Ester, M., H.-P. Kriegel, J. Sander, X. Xu, et al. (1996). "A Density-Based Algorithm for Discovering Clusters in Large Spatial Databases with Noise." In: *Kdd*. Vol. 96. 34, pp. 226–231 (cit. on p. 44).
- Faure, E. et al. (2016). "A Workflow to Process 3D+time Microscopy Images of Developing Organisms and Reconstruct Their Cell Lineage". In: *Nature Communications* 7.1 (1), p. 8674. DOI: 10.1038/ncomms9674 (cit. on p. 26).
- Förster, F., O. Medalia, N. Zauberman, W. Baumeister, and D. Fass (2005). "Retrovirus Envelope Protein Complex Structure in Situ Studied by Cryo-Electron Tomography". In: *Proceedings of the National Academy of Sciences* 102.13, pp. 4729–4734. DOI: 10.1073/pnas.0409178102 (cit. on p. 1).
- Frydenberg, M. (1990). "The Chain Graph Markov Property". In: *Scandinavian Journal of Statistics* 17.4, pp. 333–353. JSTOR: 4616181 (cit. on p. 54).
- Funke, J., L. Mais, A. Champion, N. Dye, and D. Kainmueller (2018). "A Benchmark for Epithelial Cell Tracking". In: Proceedings of the European Conference on Computer Vision (ECCV) Workshops (cit. on p. 44).
- Gidaris, S., P. Singh, and N. Komodakis (2018). "Unsupervised Representation Learning by Predicting Image Rotations". In: International Conference on Learning Representations (cit. on p. 65).
- Girshick, R. (2015). "Fast R-CNN". In: Proceedings of the IEEE International Conference on Computer Vision, pp. 1440–1448 (cit. on p. 9).
- Girshick, R., J. Donahue, T. Darrell, and J. Malik (2014). "Rich Feature Hierarchies for Accurate Object Detection and Semantic Segmentation". In: *2014 IEEE Conference on Computer Vision and Pattern Recognition*. 2014 IEEE Conference on Computer Vision and Pattern Recognition (CVPR). Columbus, OH, USA: IEEE, pp. 580–587. DOI: 10.1109/CVPR.2014.81 (cit. on p. 9).
- Gomariz, A., T. Portenier, C. Nombela-Arrieta, and O. Goksel (2021). "Probabilistic Spatial Analysis in Quantitative Microscopy with Uncertainty-Aware Cell Detection

- Using Deep Bayesian Regression of Density Maps". arXiv: 2102.11865 [cs, eess] (cit. on p. 10).
- Gurobi Optimization, LLC (2022). *Gurobi Optimizer Reference Manual* (cit. on pp. 54, 57).
- Haller, S., M. Prakash, L. Hutschenreiter, T. Pietzsch, C. Rother, F. Jug, P. Swoboda, and B. Savchynskyy (2020). "A Primal-Dual Solver for Large-Scale Tracking-by-Assignment". In: *International Conference on Artificial Intelligence and Statistics*. International Conference on Artificial Intelligence and Statistics. PMLR, pp. 2539–2549 (cit. on p. 54).
- Hamahashi, S., S. Onami, and H. Kitano (2005). "Detection of Nuclei in 4D Nomarski DIC Microscope Images of Early Caenorhabditis Elegans Embryos Using Local Image Entropy and Object Tracking". In: *BMC Bioinformatics* 6.1, p. 125. DOI: 10.1186/1471-2105-6-125 (cit. on p. 27).
- He, K., G. Gkioxari, P. Dollár, and R. Girshick (2017). "Mask R-CNN". In: *ICCV* (cit. on pp. 10, 14).
- He, K., X. Zhang, S. Ren, and J. Sun (2016). "Deep Residual Learning for Image Recognition". In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 770–778 (cit. on p. 9).
- Hirsch, P. and D. Kainmueller (2020). "An Auxiliary Task for Learning Nuclei Segmentation in 3D Microscopy Images". In: *Medical Imaging with Deep Learning*. Medical Imaging with Deep Learning. PMLR, pp. 304–321 (cit. on p. 10).
- Höfener, H., A. Homeyer, N. Weiss, J. Molin, C. F. Lundström, and H. K. Hahn (2018). "Deep Learning Nuclei Detection: A Simple Approach Can Deliver State-of-the-Art Results". In: *Computerized Medical Imaging and Graphics* 70, pp. 43–52. DOI: 10.1016/j.compmedimag.2018.08.010 (cit. on p. 10).
- Hooke, R. (1665). *Micrographia: Or, Some Physiological Descriptions of Minute Bodies Made by Magnifying Glasses*. London: J. Martyn and J. Allestry (cit. on p. 1).
- Icha, J., C. Kunath, M. Rocha-Martins, and C. Norden (2016). "Independent Modes of Ganglion Cell Translocation Ensure Correct Lamination of the Zebrafish Retina". In: *Journal of Cell Biology* 215.2, pp. 259–275. DOI: 10.1083/jcb.201604095 (cit. on p. 41).
- Jain, V. and S. Seung (2009). "Natural Image Denoising with Convolutional Networks". In: *Advances in Neural Information Processing Systems (NeurIPS)*, pp. 769–776 (cit. on p. 65).
- Jug, F., E. Levinkov, C. Blasse, E. W. Myers, and B. Andres (2016). "Moral Lineage Tracing". In: *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR). Las Vegas, NV, USA: IEEE, pp. 5926–5935. DOI: 10.1109/CVPR.2016.638 (cit. on pp. 44, 55).
- Jug, F., T. Pietzsch, D. Kainm, J. Funke, M. Kaiser, E. V. Nimwegen, C. Rother, and G. Myers (2014). "Optimal Joint Segmentation and Tracking of Escherichia Coli in the Mother Machine". In: *Bayesian and graphical Models for Biomedical Imaging*, pp. 25–36. DOI: 10.1007/978-3-319-12289-2 (cit. on pp. 44, 55).

- Kainmueller, D., F. Jug, C. Rother, and G. Myers (2014). "Active Graph Matching for Automatic Joint Segmentation and Annotation of *C. Elegans*". In: *Medical Image Computing and Computer-Assisted Intervention – MICCAI 2014*. Ed. by P. Golland, N. Hata, C. Barillot, J. Hornegger, and R. Howe. Lecture Notes in Computer Science. Cham: Springer International Publishing, pp. 81–88. DOI: 10.1007/978-3-319-10404-1_11 (cit. on p. 55).
- Kainz, P., M. Urschler, S. Schulter, P. Wohlhart, and V. Lepetit (2015). "You Should Use Regression to Detect Cells". In: *Medical Image Computing and Computer-Assisted Intervention – MICCAI 2015*. Ed. by N. Navab, J. Hornegger, W. M. Wells, and A. F. Frangi. Lecture Notes in Computer Science. Cham: Springer International Publishing, pp. 276–283. DOI: 10.1007/978-3-319-24574-4_33 (cit. on p. 10).
- Kankaanpää, P., L. Paavolainen, S. Tiitta, M. Karjalainen, J. Päivärinne, J. Nieminen, V. Marjomäki, J. Heino, and D. J. White (2012). "BioImageXD: An Open, General-Purpose and High-Throughput Image-Processing Platform". In: *Nature Methods* 9:7 (7), pp. 683–689. DOI: 10.1038/nmeth.2047 (cit. on p. 2).
- Kausler, B. X., M. Schiegg, B. Andres, M. Lindner, U. Koethe, H. Leitte, J. Wittbrodt, L. Hufnagel, and F. A. Hamprecht (2012). "A Discrete Chain Graph Model for 3d+t Cell Tracking with High Misdetetection Robustness". In: *Computer Vision – ECCV 2012*. Ed. by A. Fitzgibbon, S. Lazebnik, P. Perona, Y. Sato, and C. Schmid. Lecture Notes in Computer Science. Berlin, Heidelberg: Springer, pp. 144–157. DOI: 10.1007/978-3-642-33712-3_11 (cit. on pp. 41, 43, 45, 54).
- Keller, P. J. (2013). "Imaging Morphogenesis: Technological Advances and Biological Insights". In: *Science* 340.6137. DOI: 10.1126/science.1234168. pmid: 23744952 (cit. on p. 27).
- Keller, P. J., A. D. Schmidt, A. Santella, K. Khairy, Z. Bao, J. Wittbrodt, and E. H. K. Stelzer (2010). "Fast, High-Contrast Imaging of Animal Development with Scanned Light Sheet–Based Structured-Illumination Microscopy". In: *Nature Methods* 7:8 (8), pp. 637–642. DOI: 10.1038/nmeth.1476 (cit. on pp. 26, 27).
- Keller, P. J., A. D. Schmidt, J. Wittbrodt, and E. H. K. Stelzer (2008). "Reconstruction of Zebrafish Early Embryonic Development by Scanned Light Sheet Microscopy". In: *Science* 322.5904, pp. 1065–1069. DOI: 10.1126/science.1162493. pmid: 18845710 (cit. on pp. 26, 27).
- Khairy, K. and P. J. Keller (2011). "Reconstructing Embryonic Development". In: *genesis* 49:7, pp. 488–513. DOI: 10.1002/dvg.20698 (cit. on p. 27).
- Kingma, D. P. and J. Ba (2015). "Adam: A Method for Stochastic Optimization". In: *ICLR*. arXiv: 1412.6980 (cit. on pp. 19, 75, 77).
- Krizhevsky, A., I. Sutskever, and G. E. Hinton (2012). "ImageNet Classification with Deep Convolutional Neural Networks". In: *Advances in Neural Information Processing Systems* 25. Ed. by F. Pereira, C. J. C. Burges, L. Bottou, and K. Q. Weinberger. Curran Associates, Inc., pp. 1097–1105 (cit. on p. 9).
- Krull, A., T.-O. Buchholz, and F. Jug (2019). "Noise2Void - Learning Denoising From Single Noisy Images". In: *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. 2019 IEEE/CVF Conference on Computer Vision and Pattern

- Recognition (CVPR). Long Beach, CA, USA: IEEE, pp. 2124–2132. DOI: 10.1109/CVPR.2019.00223 (cit. on pp. 2, 65, 77).
- Krull, A., T. Vičar, M. Prakash, M. Lalit, and F. Jug (2020). “Probabilistic Noise2Void: Unsupervised Content-Aware Denoising”. In: *Frontiers in Computer Science* 2. DOI: 10.3389/fcomp.2020.00005 (cit. on p. 72).
- Laine, S., T. Karras, J. Lehtinen, and T. Aila (2019). “High-Quality Self-Supervised Deep Image Denoising”. In: *Advances in Neural Information Processing Systems* 32. Ed. by H. Wallach, H. Larochelle, A. Beygelzimer, F. d\ textquotesingle Alché-Buc, E. Fox, and R. Garnett. Curran Associates, Inc., pp. 6970–6980 (cit. on pp. 66, 68, 73).
- Larsson, G., M. Maire, and G. Shakhnarovich (2016). “Learning Representations for Automatic Colorization”. In: *Computer Vision – ECCV 2016*. Ed. by B. Leibe, J. Matas, N. Sebe, and M. Welling. Lecture Notes in Computer Science. Cham: Springer International Publishing, pp. 577–593. DOI: 10.1007/978-3-319-46493-0_35 (cit. on p. 65).
- Ledley, R. S. (1959). “Digital Electronic Computers in Biomedical Science”. In: *Science* 130.3384, pp. 1225–1234. JSTOR: 1757193 (cit. on p. 1).
- (1964). “High-Speed Automatic Analysis of Biomedical Pictures”. In: *Science* 146.3641, pp. 216–223. JSTOR: 1713812 (cit. on p. 1).
- Lee, H.-Y., J.-B. Huang, M. Singh, and M.-H. Yang (2017). “Unsupervised Representation Learning by Sorting Sequences”. In: *Proceedings of the IEEE International Conference on Computer Vision*, pp. 667–676 (cit. on p. 65).
- Lehtinen, J., J. Munkberg, J. Hasselgren, S. Laine, T. Karras, M. Aittala, and T. Aila (2018). “Noise2Noise: Learning Image Restoration without Clean Data”. In: *International Conference on Machine Learning*. International Conference on Machine Learning, pp. 2965–2974 (cit. on pp. 66, 72).
- Li, C., X. Wang, W. Liu, and L. J. Latecki (2018). “DeepMitosis: Mitosis Detection via Deep Detection, Verification and Segmentation Networks”. In: *Medical Image Analysis* 45, pp. 121–133. DOI: 10.1016/j.media.2017.12.002 (cit. on p. 5).
- Li, K., E. D. Miller, M. Chen, T. Kanade, L. E. Weiss, and P. G. Campbell (2008). “Cell Population Tracking and Lineage Construction with Spatiotemporal Context”. In: *Medical Image Analysis*. Special Issue on the 10th International Conference on Medical Imaging and Computer Assisted Intervention - MICCAI 2007 12.5, pp. 546–566. DOI: 10.1016/j.media.2008.06.001 (cit. on p. 45).
- Long, J., E. Shelhamer, and T. Darrell (2015). “Fully Convolutional Networks for Semantic Segmentation”. In: *CVPR*, p. 10 (cit. on p. 9).
- Lougee-Heimer, R. (2003). “The Common Optimization INterface for Operations Research: Promoting Open-Source Software in the Operations Research Community”. In: *IBM Journal of Research and Development* 47.1, pp. 57–66. DOI: 10.1147/rd.471.0057 (cit. on p. 54).
- Lucas, B. D. and T. Kanade (1981). “An Iterative Image Registration Technique with an Application to Stereo Vision”. In: p. 10 (cit. on p. 56).

- Luo, W., Y. Li, R. Urtasun, and R. Zemel (2016). "Understanding the Effective Receptive Field in Deep Convolutional Neural Networks". In: *Advances in Neural Information Processing Systems*. Vol. 29. Curran Associates, Inc. (cit. on p. 16).
- Magnusson, K. E. G., J. Jaldén, P. M. Gilbert, and H. M. Blau (2015). "Global Linking of Cell Tracks Using the Viterbi Algorithm". In: *IEEE Transactions on Medical Imaging* 34.4, pp. 911–929. DOI: 10.1109/TMI.2014.2370951 (cit. on pp. 44, 45).
- Malin-Mayor, C., P. Hirsch, L. Guignard, K. McDole, Y. Wan, W. C. Lemon, P. J. Keller, S. Preibisch, and J. Funke (2021). *Automated Reconstruction of Whole-Embryo Cell Lineages by Learning from Sparse Annotations*. preprint. Bioinformatics. DOI: 10.1101/2021.07.28.454016 (cit. on p. 11).
- Mandracchia, B., X. Hua, C. Guo, J. Son, T. Urner, and S. Jia (2020). "Fast and Accurate sCMOS Noise Correction for Fluorescence Microscopy". In: *Nature Communications* 11.1 (1), p. 94. DOI: 10.1038/s41467-019-13841-8 (cit. on p. 64).
- Maška, M., V. Ulman, D. Svoboda, P. Matula, P. Matula, C. Ederra, A. Urbiola, T. España, S. Venkatesan, D. M. W. Balak, P. Karas, T. Bolcková, M. Štreitová, C. Carthel, S. Coraluppi, N. Harder, K. Rohr, K. E. G. Magnusson, J. Jaldén, H. M. Blau, O. Dzyubachyk, P. Křížek, G. M. Hagen, D. Pastor-Escuredo, D. Jimenez-Carretero, M. J. Ledesma-Carbayo, A. Muñoz-Barrutia, E. Meijering, M. Kozubek, and C. Ortiz-de-Solorzano (2014). "A Benchmark for Comparison of Cell Tracking Algorithms". In: *Bioinformatics* 30.11, pp. 1609–1617. DOI: 10.1093/bioinformatics/btu080 (cit. on p. 21).
- Matas, J., O. Chum, M. Urban, and T. Pajdla (2004). "Robust Wide-Baseline Stereo from Maximally Stable Extremal Regions". In: *Image and Vision Computing*. British Machine Vision Computing 2002 22.10, pp. 761–767. DOI: 10.1016/j.imavis.2004.02.006 (cit. on p. 9).
- Matula, P., M. Maška, D. V. Sorokin, P. Matula, C. Ortiz-de-Solórzano, and M. Kozubek (2015). "Cell Tracking Accuracy Measurement Based on Comparison of Acyclic Oriented Graphs". In: *PLOS ONE* 10.12, e0144959. DOI: 10.1371/journal.pone.0144959 (cit. on p. 46).
- McDole, K., L. Guignard, F. Amat, A. Berger, G. Malandain, L. A. Royer, S. C. Turaga, K. Branson, and P. J. Keller (2018). "In Toto Imaging and Reconstruction of Post-Implantation Mouse Development at the Single-Cell Level". In: *Cell* 175.3, 859–876.e33. DOI: 10.1016/j.cell.2018.09.031 (cit. on p. 26).
- Meijering, E. (2020). "A Bird's-Eye View of Deep Learning in Bioimage Analysis". In: *Computational and Structural Biotechnology Journal* 18, pp. 2312–2325. DOI: 10.1016/j.csbj.2020.08.003 (cit. on p. 4).
- Meijering, E., A. E. Carpenter, H. Peng, F. A. Hamprecht, and J.-C. Olivo-Marin (2016). "Imagining the Future of Bioimage Analysis". In: *Nature Biotechnology* 34.12 (12), pp. 1250–1255. DOI: 10.1038/nbt.3722 (cit. on p. 1).
- Meijering, E., O. Dzyubachyk, I. Smal, and W. A. van Cappellen (2009). "Tracking in Cell and Developmental Biology". In: *Seminars in Cell & Developmental Biology*. Imaging in Cell and Developmental Biology 20.8, pp. 894–902. DOI: 10.1016/j.semcd.2009.07.004 (cit. on p. 41).

- Myers, G. (2012). "Why Bioimage Informatics Matters". In: *Nature Methods* 9.7 (7), pp. 659–660. DOI: 10.1038/nmeth.2024 (cit. on p. 1).
- Ning, F., D. Delhomme, Y. LeCun, F. Piano, L. Bottou, and P. Barbano (2005). "Toward Automatic Phenotyping of Developing Embryos from Videos". In: *IEEE Transactions on Image Processing* 14.9, pp. 1360–1371. DOI: 10.1109/TIP.2005.852470 (cit. on p. 8).
- Noroozi, M. and P. Favaro (2016). "Unsupervised Learning of Visual Representations by Solving Jigsaw Puzzles". In: *Computer Vision – ECCV 2016*. Ed. by B. Leibe, J. Matas, N. Sebe, and M. Welling. Lecture Notes in Computer Science. Cham: Springer International Publishing, pp. 69–84. DOI: 10.1007/978-3-319-46466-4_5 (cit. on p. 65).
- Padfield, D., J. Rittscher, and B. Roysam (2011). "Coupled Minimum-Cost Flow Cell Tracking for High-Throughput Quantitative Analysis". In: *Medical Image Analysis*. Special Section on IPMI 2009 15.4, pp. 650–668. DOI: 10.1016/j.media.2010.07.006 (cit. on p. 51).
- Payer, C., D. Štern, T. Neff, H. Bischof, and M. Urschler (2018). "Instance Segmentation and Tracking with Cosine Embeddings and Recurrent Hourglass Networks". In: *Medical Image Computing and Computer Assisted Intervention – MICCAI 2018*. Ed. by A. F. Frangi, J. A. Schnabel, C. Davatzikos, C. Alberola-López, and G. Fichtinger. Lecture Notes in Computer Science. Cham: Springer International Publishing, pp. 3–11. DOI: 10.1007/978-3-030-00934-2_1 (cit. on p. 44).
- Peng, H., Z. Ruan, F. Long, J. H. Simpson, and E. W. Myers (2010). "V3D Enables Real-Time 3D Visualization and Quantitative Analysis of Large-Scale Biological Image Data Sets". In: *Nature biotechnology* 28.4, pp. 348–353 (cit. on p. 13).
- Preston, K. (1976). "Computer Processing of Biomedical Images". In: *Computer* 9.5, pp. 54–68. DOI: 10.1109/C-M.1976.218587 (cit. on p. 1).
- Qu, L., F. Long, X. Liu, S. Kim, E. Myers, and H. Peng (2011). "Simultaneous Recognition and Segmentation of Cells: Application in C.Elegans". In: *Bioinformatics* 27.20, pp. 2895–2902. DOI: 10.1093/bioinformatics/btr480 (cit. on p. 55).
- Redmon, J., S. Divvala, R. Girshick, and A. Farhadi (2016). "You Only Look Once: Unified, Real-Time Object Detection". In: *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR). Las Vegas, NV, USA: IEEE, pp. 779–788. DOI: 10.1109/CVPR.2016.91 (cit. on pp. 10, 13).
- Ren, S., K. He, R. Girshick, and J. Sun (2015). "Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks". In: *Advances in Neural Information Processing Systems* 28. Ed. by C. Cortes, N. D. Lawrence, D. D. Lee, M. Sugiyama, and R. Garnett. Curran Associates, Inc., pp. 91–99 (cit. on pp. 9, 13).
- Richmond, D., D. Kainmueller, B. Glocker, C. Rother, and G. Myers (2015). "Uncertainty-Driven Forest Predictors for Vertebra Localization and Segmentation". In: *Medical Image Computing and Computer-Assisted Intervention – MICCAI 2015*. Ed. by N. Navab, J. Hornegger, W. M. Wells, and A. Frangi. Lecture Notes in Computer Science. Cham: Springer International Publishing, pp. 653–660. DOI: 10.1007/978-3-319-24553-9_80 (cit. on p. 12).

- Ronneberger, O., P. Fischer, and T. Brox (2015). "U-Net: Convolutional Networks for Biomedical Image Segmentation". In: *Medical Image Computing and Computer-Assisted Intervention – MICCAI 2015*. Ed. by N. Navab, J. Hornegger, W. M. Wells, and A. F. Frangi. Lecture Notes in Computer Science. Cham: Springer International Publishing, pp. 234–241. DOI: 10.1007/978-3-319-24574-4_28 (cit. on pp. 4, 9, 13, 16, 74).
- Roth, S. and M. Black (2005). "Fields of Experts: A Framework for Learning Image Priors". In: *2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05)*. 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05). Vol. 2, 860–867 vol. 2. DOI: 10.1109/CVPR.2005.160 (cit. on p. 65).
- Rust, M. J., M. Bates, and X. Zhuang (2006). "Stochastic Optical Reconstruction Microscopy (STORM) Provides Sub-Diffraction-Limit Image Resolution". In: *Nature methods* 3.10, pp. 793–795. DOI: 10.1038/nmeth929. pmid: 16896339 (cit. on p. 14).
- Saalfeld, S., A. Cardona, V. Hartenstein, and P. Tomančák (2009). "CATMAID: Collaborative Annotation Toolkit for Massive Amounts of Image Data". In: *Bioinformatics* 25.15, pp. 1984–1986. DOI: 10.1093/bioinformatics/btp266 (cit. on p. 13).
- Sbalzarini, I. F. (2016). "Seeing Is Believing: Quantifying Is Convincing: Computational Image Analysis in Biology". In: *Focus on Bio-Image Informatics*. Ed. by W. H. De Vos, S. Munck, and J.-P. Timmermans. Advances in Anatomy, Embryology and Cell Biology. Cham: Springer International Publishing, pp. 1–39. DOI: 10.1007/978-3-319-28549-8_1 (cit. on p. 1).
- Schiegg, M., P. Hanslovsky, B. X. Kausler, L. Hufnagel, and F. A. Hamprecht (2013). "Conservation Tracking". In: *Proceedings of the IEEE International Conference on Computer Vision*, pp. 2928–2935 (cit. on pp. 12, 43, 45, 55).
- Schindelin, J., I. Arganda-Carreras, E. Frise, V. Kaynig, M. Longair, T. Pietzsch, S. Preibisch, C. Rueden, S. Saalfeld, B. Schmid, J.-Y. Tinevez, D. J. White, V. Hartenstein, K. Eliceiri, P. Tomancak, and A. Cardona (2012). "Fiji: An Open-Source Platform for Biological-Image Analysis". In: *Nature Methods* 9.7 (7), pp. 676–682. DOI: 10.1038/nmeth.2019 (cit. on pp. 2, 13, 42, 64).
- Schmidt, U. and S. Roth (2014). "Shrinkage Fields for Effective Image Restoration". In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 2774–2781 (cit. on p. 65).
- Schmidt, U., M. Weigert, C. Broaddus, and G. Myers (2018). "Cell Detection with Star-Convex Polygons". In: *Medical Image Computing and Computer Assisted Intervention – MICCAI 2018*. Ed. by A. F. Frangi, J. A. Schnabel, C. Davatzikos, C. Alberola-López, and G. Fichtinger. Lecture Notes in Computer Science. Cham: Springer International Publishing, pp. 265–273. DOI: 10.1007/978-3-030-00934-2_30 (cit. on pp. 2, 5, 13, 65).
- Schmitz, C., B. S. Eastwood, S. J. Tappan, J. R. Glaser, D. A. Peterson, and P. R. Hof (2014). "Current Automated 3D Cell Detection Methods Are Not a Suitable

- Replacement for Manual Stereologic Cell Counting". In: *Frontiers in Neuroanatomy* 8. DOI: 10.3389/fnana.2014.00027 (cit. on p. 7).
- Schneider, C. A., W. S. Rasband, and K. W. Eliceiri (2012). "NIH Image to ImageJ: 25 Years of Image Analysis". In: *Nature Methods* 9.7, pp. 671–675. DOI: 10.1038/nmeth.2089 (cit. on p. 2).
- Sirinukunwattana, K., S. E. A. Raza, Y.-W. Tsang, D. R. J. Snead, I. A. Cree, and N. M. Rajpoot (2016). "Locality Sensitive Deep Learning for Detection and Classification of Nuclei in Routine Colon Cancer Histology Images". In: *IEEE Transactions on Medical Imaging* 35.5, pp. 1196–1206. DOI: 10.1109/TMI.2016.2525803 (cit. on p. 10).
- Sjöström, P. J., B. R. Frydel, and L. U. Wahlberg (1999). "Artificial Neural Network-Aided Image Analysis System for Cell Counting". In: *Cytometry* 36.1, pp. 18–26. DOI: 10.1002/(SICI)1097-0320(19990501)36:1<18::AID-CYT03>3.0.CO;2-J (cit. on p. 8).
- Smal, I., E. Meijering, K. Draegestein, N. Galjart, I. Grigoriev, and A. Akhmanova (2008). "Multiple Object Tracking in Molecular Bioimaging by Rao-Blackwellized Marginal Particle Filtering". In: *Medical Image Analysis*, p. 14 (cit. on p. 45).
- Sofroniew, N., T. Lambert, K. Evans, J. Nunez-Iglesias, P. Winston, K. Yamauchi, G. Bokota, A. C. Solak, Ziyangczi, G. Buckley, D. D. Pop, T. Tung, Hector, J. Freeman, M. Bussonnier, P. Boone, V. Hilsenstein, A. R. Lowe, L. Royer, H. Har-Gil, M. Kittisopikul, S. Axelrod, Alisterburt, A. Patil, A. McGovern, A. Rokem, Bryant, and C. Gohlke (2020). *Napari/Napari: 0.4.2rc0*. Version v0.4.2rc0. Zenodo. DOI: 10.5281/ZENODO.4294808 (cit. on pp. 13, 14, 78).
- Sommer, C., C. Straehle, U. Köthe, and F. A. Hamprecht (2011). "Ilastik: Interactive Learning and Segmentation Toolkit". In: *2011 IEEE International Symposium on Biomedical Imaging: From Nano to Macro*. 2011 IEEE International Symposium on Biomedical Imaging: From Nano to Macro, pp. 230–233. DOI: 10.1109/ISBI.2011.5872394 (cit. on pp. 11, 13, 54, 59).
- Speiser, A., L.-R. Müller, U. Matti, C. J. Obara, W. R. Legant, J. Ries, J. H. Macke, and S. C. Turaga (2020). "Teaching Deep Neural Networks to Localize Single Molecules for Super-Resolution Microscopy". arXiv: 1907.00770 [cs, eess, stat] (cit. on p. 14).
- Stapel, L. C., B. Lombardot, C. Broaddus, D. Kainmueller, F. Jug, E. W. Myers, and N. L. Vastenhouw (2016). "Automated Detection and Quantification of Single RNAs at Cellular Resolution in Zebrafish Embryos". In: *Development* 143.3, pp. 540–546. DOI: 10.1242/dev.128918 (cit. on p. 12).
- Stegmaier, J., F. Amat, W. C. Lemon, K. McDole, Y. Wan, G. Teodoro, R. Mikut, and P. J. Keller (2016). "Real-Time Three-Dimensional Cell Segmentation in Large-Scale Microscopy Data of Developing Embryos". In: *Developmental Cell* 36.2, pp. 225–240. DOI: 10.1016/j.devcel.2015.12.028. pmid: 26812020 (cit. on p. 64).
- Stewart, R., M. Andriluka, and A. Y. Ng (2016). "End-To-End People Detection in Crowded Scenes". In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 2325–2333 (cit. on p. 11).

- Strigl, D., K. Kofler, and S. Podlipnig (2010). "Performance and Scalability of GPU-Based Convolutional Neural Networks". In: *2010 18th Euromicro Conference on Parallel, Distributed and Network-based Processing*. 2010 18th Euromicro Conference on Parallel, Distributed and Network-based Processing, pp. 317–324. DOI: 10.1109/PDP.2010.43 (cit. on p. 9).
- Sugawara, K., C. Cevrim, and M. Averof (2021). "Tracking Cell Lineages in 3D by Incremental Deep Learning". In: *bioRxiv*, p. 2021.02.26.432552. DOI: 10.1101/2021.02.26.432552 (cit. on p. 10).
- Sulston, J., E. Schierenberg, J. White, and J. Thomson (1983). "The Embryonic Cell Lineage of the Nematode *Caenorhabditis Elegans*". In: *Developmental Biology* 100.1, pp. 64–119. DOI: 10.1016/0012-1606(83)90201-4 (cit. on p. 42).
- Sun, D., S. Roth, and M. J. Black (2010). "Secrets of Optical Flow Estimation and Their Principles". In: *2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*. 2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, pp. 2432–2439. DOI: 10.1109/CVPR.2010.5539939 (cit. on p. 55).
- Swoger, J., P. Verveer, K. Greger, J. Huiskens, and E. H. K. Stelzer (2007). "Multi-View Image Fusion Improves Resolution in Three-Dimensional Microscopy". In: *Optics Express* 15.13, pp. 8029–8042. DOI: 10.1364/OE.15.008029 (cit. on p. 1).
- Thierbach, K., P.-L. Bazin, F. Gavriilidis, E. Kirilina, C. Jäger, M. Morawski, S. Geyer, N. Weiskopf, and N. Scherf (2018). "Deep Learning Meets Topology-preserving Active Contours: Towards Scalable Quantitative Histology of Cortical Cytoarchitecture". In: *bioRxiv*, p. 297689. DOI: 10.1101/297689 (cit. on p. 10).
- Thirion, J.-. (1998). "Image Matching as a Diffusion Process: An Analogy with Maxwell's Demons". In: *Medical Image Analysis* 2.3, pp. 243–260. DOI: 10.1016/S1361-8415(98)80022-4 (cit. on p. 55).
- Tinevez, J.-Y., N. Perry, J. Schindelin, G. M. Hoopes, G. D. Reynolds, E. Laplantine, S. Y. Bednarek, S. L. Shorte, and K. W. Eliceiri (2017). "TrackMate: An Open and Extensible Platform for Single-Particle Tracking". In: *Methods. Image Processing for Biologists* 115, pp. 80–90. DOI: 10.1016/j.ymeth.2016.09.016 (cit. on p. 42).
- Uetz, R. and S. Behnke (2009). "Large-Scale Object Recognition with CUDA-accelerated Hierarchical Neural Networks". In: *2009 IEEE International Conference on Intelligent Computing and Intelligent Systems*. 2009 IEEE International Conference on Intelligent Computing and Intelligent Systems. Vol. 1, pp. 536–541. DOI: 10.1109/ICICISYS.2009.5357786 (cit. on p. 9).
- Uijlings, J. R. R., K. E. A. van de Sande, T. Gevers, and A. W. M. Smeulders (2013). "Selective Search for Object Recognition". In: *International Journal of Computer Vision* 104.2 (2), pp. 154–171. DOI: 10.1007/s11263-013-0620-5 (cit. on p. 9).
- Ulman, V. et al. (2017). "An Objective Comparison of Cell-Tracking Algorithms". In: *Nature Methods* 14.12 (12), pp. 1141–1152. DOI: 10.1038/nmeth.4473 (cit. on pp. 21, 64, 76).

- Ulyanov, D., A. Vedaldi, and V. Lempitsky (2018). "Deep Image Prior". In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 9446–9454 (cit. on p. 73).
- Van der Walt, S., J. L. Schönberger, J. Nunez-Iglesias, F. Boulogne, J. D. Warner, N. Yager, E. Gouillart, and T. Yu (2014). "Scikit-Image: Image Processing in Python". In: *PeerJ* 2, e453. DOI: 10.7717/peerj.453 (cit. on pp. 19, 64).
- Verveer, P. J., J. Swoger, F. Pampaloni, K. Greger, M. Marcello, and E. H. K. Stelzer (2007). "High-Resolution Three-Dimensional Imaging of Large Specimens with Light Sheet-Based Microscopy". In: *Nature Methods* 4.4 (4), pp. 311–313. DOI: 10.1038/nmeth1017 (cit. on p. 1).
- Vincent, P., H. Larochelle, I. Lajoie, Y. Bengio, and P.-A. Manzagol (2010). "Stacked Denoising Autoencoders: Learning Useful Representations in a Deep Network with a Local Denoising Criterion". In: *Journal of machine learning research* 11.12, p. 38 (cit. on p. 65).
- Virtanen, P. et al. (2020). "SciPy 1.0: Fundamental Algorithms for Scientific Computing in Python". In: *Nature Methods* 17.3 (3), pp. 261–272. DOI: 10.1038/s41592-019-0686-2 (cit. on p. 17).
- Waithe, D., M. Hailstone, M. K. Lalwani, R. Parton, L. Yang, R. Patient, C. Eggeling, and I. Davis (2016). "3-D Density Kernel Estimation for Counting in Microscopy Image Volumes Using 3-D Image Filters and Random Decision Trees". In: *Computer Vision – ECCV 2016 Workshops*. Ed. by G. Hua and H. Jégou. Lecture Notes in Computer Science. Cham: Springer International Publishing, pp. 244–255. DOI: 10.1007/978-3-319-46604-0_18 (cit. on p. 10).
- Wang, Z., A. Bovik, H. Sheikh, and E. Simoncelli (2004). "Image Quality Assessment: From Error Visibility to Structural Similarity". In: *IEEE Transactions on Image Processing* 13.4, pp. 600–612. DOI: 10.1109/TIP.2003.819861 (cit. on pp. 76, 77).
- Weigert, M. (2019). "Computational Methods for Visualization, Simulation, and Restoration of Fluorescence Microscopy Data" (cit. on pp. 64, 65).
- Weigert, M., U. Schmidt, T. Boothe, A. Müller, A. Dibrov, A. Jain, B. Wilhelm, D. Schmidt, C. Broaddus, S. Culley, M. Rocha-Martins, F. Segovia-Miranda, C. Norden, R. Henriques, M. Zerial, M. Solimena, J. Rink, P. Tomancak, L. Royer, F. Jug, and E. W. Myers (2018). "Content-Aware Image Restoration: Pushing the Limits of Fluorescence Microscopy". In: *Nature Methods* 15.12, pp. 1090–1097. DOI: 10.1038/s41592-018-0216-7 (cit. on pp. 2, 5, 12, 64).
- Weigert, M., U. Schmidt, R. Haase, K. Sugawara, and G. Myers (2020). "Star-Convex Polyhedra for 3D Object Detection and Segmentation in Microscopy". In: *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pp. 3666–3673 (cit. on pp. 2, 5, 13).
- Willert, C. E. and M. Gharib (1991). "Digital Particle Image Velocimetry". In: *Experiments in Fluids* 10.4, pp. 181–193. DOI: 10.1007/BF00190388 (cit. on p. 56).
- Wolff, C., J.-Y. Tinevez, T. Pietzsch, E. Stamataki, B. Harich, L. Guignard, S. Preibisch, S. Shorte, P. J. Keller, P. Tomancak, and A. Pavlopoulos (2018). "Multi-View

- Light-Sheet Imaging and Tracking with the MaMuT Software Reveals the Cell Lineage of a Direct Developing Arthropod Limb". In: *eLife* 7. Ed. by A. Sánchez Alvarado, e34410. DOI: 10.7554/eLife.34410 (cit. on pp. 13, 42).
- Wu, Q., F. Merchant, and K. Castleman (2008). *Microscope Image Processing*. Elsevier (cit. on p. 64).
- Xie, J., L. Xu, and E. Chen (2012). "Image Denoising and Inpainting with Deep Neural Networks". In: *Advances in Neural Information Processing Systems* 25. Ed. by F. Pereira, C. J. C. Burges, L. Bottou, and K. Q. Weinberger. Curran Associates, Inc., pp. 341–349 (cit. on p. 65).
- Xie, Y., X. Kong, F. Xing, F. Liu, H. Su, and L. Yang (2015). "Deep Voting: A Robust Approach Toward Nucleus Localization in Microscopy Images". In: *Medical image computing and computer-assisted intervention : MICCAI ... International Conference on Medical Image Computing and Computer-Assisted Intervention* 9351, pp. 374–382. DOI: 10.1007/978-3-319-24574-4_45. PMID: 28083567 (cit. on p. 10).
- Xie, Y., F. Xing, X. Shi, X. Kong, H. Su, and L. Yang (2018). "Efficient and Robust Cell Detection: A Structured Regression Approach". In: *Medical image analysis* 44, pp. 245–254. DOI: 10.1016/j.media.2017.07.003. PMID: 28797548 (cit. on p. 10).
- Yasuda, T., H. Bannai, S. Onami, S. Miyano, and H. Kitano (1999). "Towards Automatic Construction of Cell-Lineage of *C. Elegans* from Nomarski DIC Microscope Images". In: *Genome Informatics* 10, pp. 144–154. DOI: 10.11234/gi1990.10.144 (cit. on p. 27).
- Zhang, K., W. Zuo, Y. Chen, D. Meng, and L. Zhang (2017). "Beyond a Gaussian Denoiser: Residual Learning of Deep CNN for Image Denoising". In: *IEEE Transactions on Image Processing* 26.7, pp. 3142–3155. DOI: 10.1109/TIP.2017.2662206. arXiv: 1608.03981 (cit. on p. 65).
- Zhang, R., P. Isola, and A. A. Efros (2016). "Colorful Image Colorization". In: *Computer Vision – ECCV 2016*. Ed. by B. Leibe, J. Matas, N. Sebe, and M. Welling. Lecture Notes in Computer Science. Cham: Springer International Publishing, pp. 649–666. DOI: 10.1007/978-3-319-46487-9_40 (cit. on p. 65).
- Zhao, Z., L. Yang, H. Zheng, I. H. Guldner, S. Zhang, and D. Z. Chen (2018). "Deep Learning Based Instance Segmentation in 3D Biomedical Images Using Weak Annotation". In: *Medical Image Computing and Computer Assisted Intervention – MICCAI 2018*. Ed. by A. F. Frangi, J. A. Schnabel, C. Davatzikos, C. Alberola-López, and G. Fichtinger. Lecture Notes in Computer Science. Cham: Springer International Publishing, pp. 352–360. DOI: 10.1007/978-3-030-00937-3_41 (cit. on p. 10).