UNIVERSIDADE FEDERAL DO RIO GRANDE DO SUL
INSTITUTO DE INFORMÁTICA
PROGRAMA DE PÓS-GRADUAÇÃO EM COMPUTAÇÃO

LUCAS NEDEL KIRSTEN

Detecting and Tracking Cells in Microscopic Images using Oriented Representations

Thesis presented in partial fulfillment of the requirements for the degree of Master of Computer Science

Advisor: Prof. Dr. Cláudio Rosito Jung

Porto Alegre
May 2023

CIP — CATALOGING-IN-PUBLICATION

Kirsten, Lucas Nedel

Detecting and Tracking Cells in Microscopic Images using Oriented Representations / Lucas Nedel Kirsten. – Porto Alegre: PPGC da UFRGS, 2023.

68 f.: il.

Thesis (Master) – Universidade Federal do Rio Grande do Sul. Programa de Pós-Graduação em Computação, Porto Alegre, BR–RS, 2023. Advisor: Cláudio Rosito Jung.

1. Oriented object detection. 2. Cell detection. 3. Cell tracking. I. Jung, Cláudio Rosito. II. Título.

“If a machine is expected to be infallible,
it cannot also be intelligent.”
— Alan Turing

AGRADECIMENTOS

Agradeço aos meus pais, Édison e Cristina, por todo o apoio e dedicação.

À minha irmã, Camila, e a toda minha família por sempre acreditarem em minha capacidade.

À minha namorada, Diandra, por sempre estar ao meu lado.

Aos amigos e colegas, pelos momentos de descontração e trocas de conhecimento.

Ao professor Cláudio Jung por toda sua dedicação.

ABSTRACT

Cell detection and tracking are paramount for bio-analysis. Recent approaches rely on the tracking by model evolution paradigm, which usually consists of training end-to-end deep learning models to detect and track the cells on the frames with promising results. However, such methods require extensive amounts of annotated data, which is time-consuming and often requires specialized annotators. This work proposes a new approach based on the classical tracking-by-detection paradigm that alleviates the requirement of annotated data. More precisely, it approximates the cell shapes as oriented ellipses and then uses general-purpose oriented object detectors to identify the cells in each frame. We then rely on a global data association algorithm that explores temporal cell similarity using probability distance metrics, considering that the ellipses relate to two-dimensional Gaussian distributions. Our results show that our method can achieve detection and tracking results competitively with SOTA techniques that require considerably more extensive data annotation. Our code is available at: <https://github.com/LucasKirsten/Deep-Cell-Tracking-EBB>.

Keywords: Oriented object detection. Cell detection. Cell tracking.

Detecção e Rastreamento de Células em Imagens Microscópicas usando Representação Orientada

RESUMO

Detecção e rastreamento de células são fundamentais para a bioanálise. Abordagens recentes se baseiam no paradigma de rastreamento por evolução de modelo, que geralmente consiste em treinar modelos de aprendizado profundo de ponta a ponta para detectar e rastrear as células nos quadros obtendo resultados promissores. No entanto, tais métodos requerem grandes quantidades de dados anotados, que são demorados para serem obtidos e muitas vezes requerem anotadores especializados. Este trabalho propõe uma nova abordagem baseada no paradigma clássico de rastreamento por detecção que alivia a necessidade de dados anotados. Mais precisamente, ela aproxima as formas das células como elipses orientadas e, em seguida, usa detectores de objetos orientados de propósito geral para identificar as células em cada quadro. Utilizamos então um algoritmo de associação global de objetos que explora a similaridade temporal das células usando métricas de distância de probabilidade, considerando que as elipses se referem a distribuições gaussianas bidimensionais. Nossos resultados mostram que nosso método pode alcançar resultados de detecção e rastreamento competitivos com técnicas estado-da-arte que exigem consideravelmente mais anotações de dados. Nosso código está disponível em: <https://github.com/LucasKirsten/Deep-Cell-Tracking-EBB>.

Palavras-chave: Detecção de objetos orientados. Detecção de células. Rastreamento de células.

LIST OF ABBREVIATIONS AND ACRONYMS

AP	Average Precision
BB	Bounding Box
BLOB	Cell tracking-by-detection method by Akram et al. (2016)
CSL	Circular Smooth Label model by Yang, Yan and He (2020)
CTC	Cell Tracking Challenge (MAŠKA et al., 2014)
CPN	Cell Proposal Network tracking-by-detection method by Akram et al. (2017)
CNN	Convolutional Neural Network
DCL	Densely Coded Labels model by Yang et al. (2021)
DRL	Deep Reinforcement Learning tracking-by-detection method by Wang et al. (2020a)
EBB	Elliptical Bounding Box
EPFL	Cell tracking-by-detection method by Türetken et al. (2015)
FP	False Positive
FPN	Feature Pyramid Network
FN	False Negative
F1	F1-score
GBB	Gaussian Bounding Box
GOWT1	Fluo-N2DH-GOWT1 dataset from the CTC (MAŠKA et al., 2014)
GT	Ground-Truth
GPU	Graphics Processing Unit
HBB	Horizontal Bounding Box
HEID	Cell tracking-by-detection method by Türetken et al. (2015)
HeLa	Fluo-N2DH-HeLa dataset from the CTC (MAŠKA et al., 2014)
IDF1	ID-MEASURE (RISTANI et al., 2016) metric computed with F1-score
IDP	ID-MEASURE (RISTANI et al., 2016) metric computed with Precision

IDR	ID-MEASURE (RISTANI et al., 2016) metric computed with Recall
ILP	Integer Linear Programming
IoU	Intersection over Union
KTH	Cell tracking-by-detection method by Magnusson and Jaldén (2012)
MOTA	CLEAR-MOT (BERNARDIN; STIEFELHAGEN, 2008; MILAN et al., 2016) metric
NMS	Non-Maximum Suppression
OBB	Oriented Bounding Box
OCD	Oriented Cell Dataset
P	Precision
RetinaNet	Oriented object detector by Lin et al. (2017b)
RoI	Region of Interest
RPN	Region Proposal Network
R3Det	Oriented object detector by Yang et al. (2021)
R^2CNN	Oriented object detector by Jiang et al. (2017)
RSDet	Oriented object detector by Qian et al. (2019)
R	Recall
SOTA	State-Of-The-Art
SHAP	Shapley Additive Explanations (LUNDBERG; LEE, 2017; LUNDBERG et al., 2018)
ST-TCV	Cell tracking-by-detection method by Boukari and Makrogiannis (2018)
TP	True Positive
U373	PhC-C2DH-U373 dataset from the CTC (MAŠKA et al., 2014)
UFRGS	Universidade Federal do Rio Grande do Sul

LIST OF SYMBOLS

px	Pixel-size
det	Determinant operation
ln	Natural logarithm
$\boldsymbol{\mu}$	Mean vector
Σ	Covariance matrix
$\mathcal{N}$	Normal distribution
$\boldsymbol{\rho}$	Vector containing the likelihood value for each hypothesis
$\boldsymbol{C}$	Binary matrix containing the constraints for all possible hypotheses
B_D	Bhattacharyya distance
H_D	Helinger distance
W	Width
H	Height
θ	Rotation angle
Δt	Frame capture time
α	Computed object detector precision for a given data set
τ_s	Score threshold
τ_h	Overlap threshold used for suppressing detection
τ_o	Overlap threshold used to associate detections of subsequent frames
t_{th}	Time threshold
τ_{FP}	False positive score threshold
P_{link}	Link hypothesis likelihood
P_{mit}	Mitoses hypothesis likelihood
P_{FP}	False positive hypothesis likelihood
P_{cplt}	Completeness hypothesis likelihood

λ_{link} Free-parameter that controls the link hypothesis likelihood exponential decay

λ_{mit} Free-parameter that controls the mitoses hypothesis likelihood exponential decay

LIST OF FIGURES

LIST OF TABLES

CONTENTS

1 INTRODUCTION

1.1 Motivations

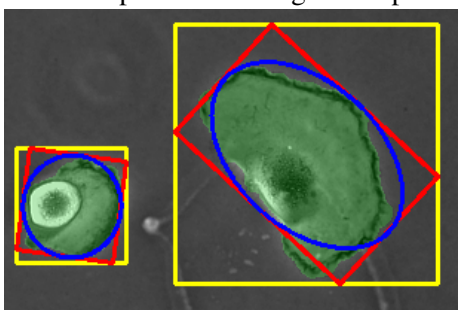
Detection and tracking of living cells in microscopy images is a crucial task required in many biomedical applications, such as cell growth, migration, invasion, morphological changes, and changes in the localization of molecules within cells (SYED et al., 2008; LEITE; CESTARI; CESTARI, 2015; GIUSEPPE et al., 2019; GRADECI et al., 2020). The sheer amount of data produced by high-throughput microscopy imaging imposes an analytical challenge for science researchers, which can only be overcome with the appropriate computational tools.

As with several other computer vision tasks, the state-of-the-art (SOTA) for cell detection and tracking is based on deep learning approaches (HAYASHIDA; NISHIMURA; BISE, 2022; EMAMI; SEDAEI; FERDOUSI, 2021). These techniques typically require manual cell annotations for training and evaluating the models, and the annotation format has a significant impact on both the time devoted to image labeling and the complexity of the network itself. The most traditional object representation refers to using horizontal bounding boxes (HBBs, a.k.a. BBs) to detect objects in a scene. Despite being very simple to annotate, this representation is not adequate when dealing with oriented elongated objects, since the HBB may contain large portions of the background or other objects in clutter scenarios. On the other hand, segmenting each object provides a fine-grained representation of the shape, but it is a tedious and time-consuming task. Moreover, applications that use multiple cell lineages from different sources (e.g., microscope, cell type) may require several rounds of labeling data and retraining the models (ULMAN et al., 2017). In these cases, a fast and efficient method for quickly labeling the data is crucial for the application continuity, since it is usually the most time-consuming step.

In the context of cell detection and tracking, knowing the complete shape representation might not be needed, while it can impose a real challenge in cases where the cell contour is highly uncertain (e.g., when there is low contrast between the cells and the background). Furthermore, methods used for individually segmenting the cell masks usually require more complex and computationally expensive algorithms, since they are typically developed in a two-step manner (either detecting and then segmenting (AKRAM et al., 2017; HE et al., 2017), or segmenting and then splitting the masks (BISE; YIN; KANADE, 2011; RONNEBERGER; FISCHER; BROX, 2015; BENSCH; RONNEBERGER, 2015;

GUPTA et al., 2019; WANG et al., 2020a)). Oriented bounding boxes (OBBs, a.k.a. rotated bounding boxes) are an intermediate representation between segmentation masks and HBBs with a good compromise between simplicity and completeness. However, the presence of roughly circular cells imposes angular ambiguity on its representation, since its OBB representation will be a square rotated at any angle (see the left cell on Figure 1.1).

Figure 1.1 – Comparison of different types of annotations on the Cell Tracking Challenge (MAŠKA et al., 2014) dataset. In green is the full segmentation mask, in yellow is the Horizontal Bounding Box representation, in red is the Oriented Bounding Box representation, and in blue is the Elliptical Bounding Box representation.



Source: Modified from Maška et al. (2014).

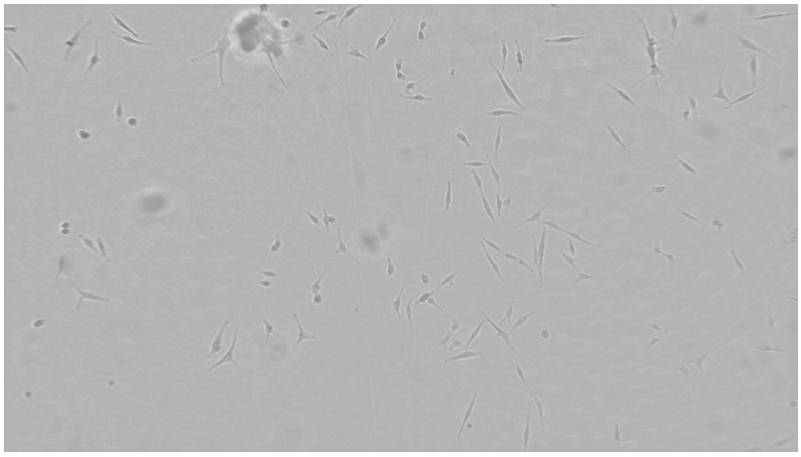
1.2 Objectives and contributions

In this work, we study the use of oriented representation for cell detection in two scenarios: (i) on a private OBB annotated cell dataset (composed by different cell types and lineages); and (ii) on generic datasets provided by the Cell Tracking Challenge (CTC) (MAŠKA et al., 2014) for approximating segmentation masks. The private cell dataset is composed mainly by glioblastoma cells, which present low contrast with respect to the background, and images typically present artifacts similar to the cells (as illustrated in Figure 1.2) that makes it very difficult to segment the individual cells consistently. Furthermore, the shape and size of glioblastoma cells can vary considerably, but they have a mostly elongated shape. As such, using HBBs for detecting them is not a good choice, since the HBB of one cell may not capture the actual shape/elongation, while also can contain the neighboring object and possibly large portions of the background. For the open-source CTC images, we advocate using elliptical bounding boxes (EBBs), which can be directly derived from OBBs and can capture oriented cells while mitigating the angular ambiguity for circular objects. Figure 1.1 shows a comparison of HBBs, OBBs, segmentation masks, and EBBs for two different cells: the left one is roughly circular, and the right one is oriented. The EBB representation, shown in blue, presents a good

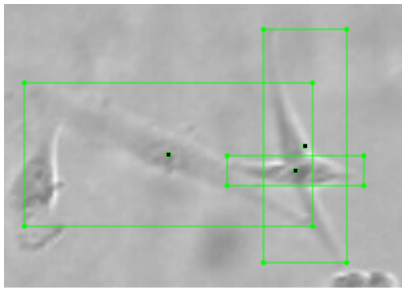
fit in both examples. As an additional advantage, the orientation/shape of detected cells represented as EBBs can be explored in tracking-by-detection approaches to provide a better spatio-temporal association when time-lapse sequences are used, while weakly-supervised segmentation methods such as the one proposed by Kulharia et al. (2020) can be coupled to the detected OBBs/EBBs to obtain a more detailed representation of the cell shape.

Figure 1.2 – Example of (a) glioblastoma cell image, and comparison of (b) HBBs and (c) OBBs annotations.

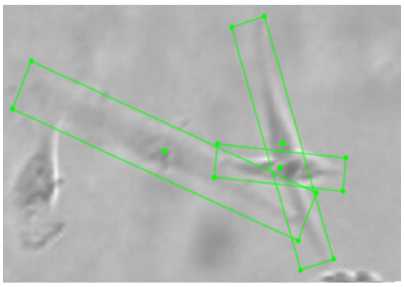
(a) Glioblastoma cell image.



(b) HBB representation.



(c) OBB representation.



Source: The authors.

Finally, we propose a cell tracking-by-detection method that uses a general-purpose deep learning model to detect the cells as OBBs, then convert them to EBBs to fit the cell shapes better. For the tracking part, we based our solution on the work of Bise, Yin and Kanade (2011), which describes an unsupervised (i.e., no tracking label is necessary) long-term global data association algorithm. We adapted their algorithm to rely only on the detection information provided by the object detector model (e.g., position, confidence score), eliminating the necessity of extracting object features from the images (e.g., histograms) and, hence, allowing the method to be used in broader spectrum of applications. We also propose a method for computing the overlap degree between detections based on probabilistic similarity measures. More precisely, we describe the EBBs as Gaussian

distributions and use the Helinger distance (HELLINGER, 1909) to compute the overlap between detections. This method solves the OBB orientation ambiguity, and allows a simple and efficient way to compute the similarity between EBBs. We explored this formulation both in suppressing multiple detections related to the same object (i.e., in non-maximum suppression) and to associate detections in subsequent frames to generate a tracklet. It is important to note that our method only requires OBB cell annotations for isolated frames, and no tracking annotation involving temporal sequences is needed.

This work is organized as follows:

- **Section 2** provides a review of the current literature related to general purpose object detection, and then specific to cell detection, segmentation and tracking.
- **Section 3** describes our complete method of cell tracking-by-detection using the EBB representation to approximate the cell shapes.
- **Section 4** describes the conducted experiments regarding the used data sets and its manipulation, the adopted evaluation protocol, the tests with general purpose OBB detectors, and the parameters value definition used in our method.
- **Section 5** presents the results of our experiments, and a sensitive analyses of our method to its parameters choice.
- **Section 6** presents our final conclusions and future works.

2 RELATED WORK

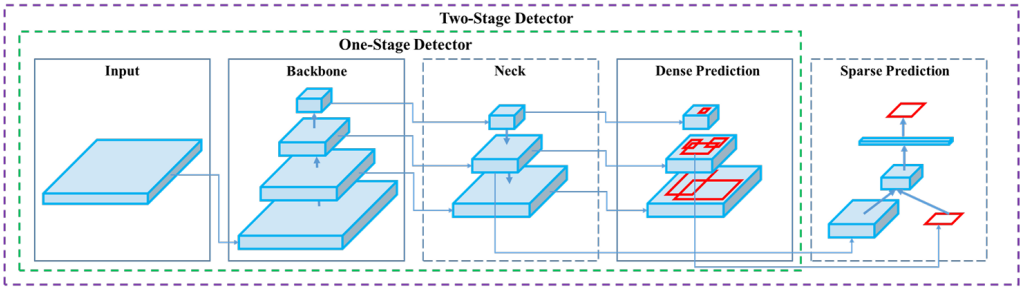
In this section, we review the current literature regarding cell detection and tracking, along with a discussion about evaluation metrics. Since most of the works developed in this area are usually tailored solutions based on general-purpose solutions, we start by presenting popular and SOTA works for detecting generic objects. Next, we proceed to describe the specific task of cell detection and segmentation, as well as other proposals for cell shape approximation. Then, we review the cell tracking paradigm, and finally we discuss current metrics used to evaluate general-purpose object detectors and trackers.

2.1 General purpose object detection with oriented bounding boxes

Despite the existence of several methods tailored for cell detection and tracking, they are mostly specialized versions of algorithms for generic object detection/tracking. General purpose object detection with HBBs is already a well-consolidated problem in computer vision with several improvements in the past years, as noted in the recent survey paper by Zaidi et al. (2022). However, the literature regarding object detection with OBBs is more recent and scarce, since moving from HBBs to OBBs adds some challenges, such as the ambiguous parametrization of OBBs, the difficulty in regressing angular information, and the adaptation of anchor-based methods (YANG et al., 2021a). Furthermore, datasets containing annotated OBBs are mostly restricted to niche applications, such as text localization (e.g., ICDAR2015 (KARATZAS et al., 2015), MLT2017 (NAYEF et al., 2017)) or object detection in aerial/satellite images (e.g., HRSC2016 (LIU et al., 2017), DOTA (XIA et al., 2018)). Nevertheless, as the HBB counterpart, OBB object detection can be mainly performed using one- or two-stage methods. Figure 2.1 illustrates the main components of each of these methods. First, the input image is fed to a backbone model, which will extract features from it. Popular choices of backbones are VGG (SIMONYAN; ZISSERMAN, 2014), ResNet (HE et al., 2016), and AlexNet (DENG et al., 2009), to mention a few. Necks connect the extracted features to the head layers (LIU et al., 2020), and multiscale fusion through Feature Pyramid Networks (FPNs) (LIN et al., 2017a), and BiDirectional FPNs (TAN; PANG; LE, 2020) are becoming popular. Finally, the head outputs the dense or sparse predictions of the object representation parameters (e.g., width, height, x-center, y-center, and angle for OBBs) and the corresponding categories.

In two-stage methods, the first stage creates OBB proposals, and the second predicts

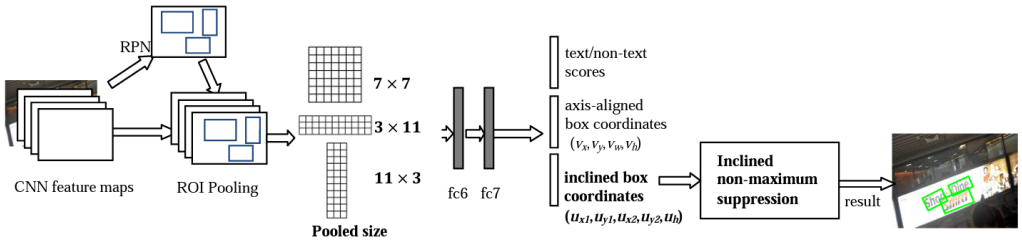
Figure 2.1 – Illustration of the main components in one- and two-stage object detectors.



Source: Bochkovskiy, Wang and Liao (2020).

the class-related confidence for each proposal and refines its shape. As examples, He and Lau (2015) proposed to extract oriented proposals from 2D Gaussian distributions of feature maps, while Ding et al. (2019) proposed a Region of Interest (RoI) transformer that adjusts horizontal RoIs to oriented counterparts. Xia et al. (2018) adapted the popular Faster-RCNN method (REN et al., 2016) to generate and evaluate OBB proposals in the context of object detection in aerial/satellite images. To avoid developing specific pooling strategies for OBBs, R^{2}CNN (JIANG et al., 2017) explores three horizontal RoI-pooling layers on the enclosing box of the OBB, and develop a regression strategy for generating OBBs as final representations, as illustrated in Figure 2.2.

Figure 2.2 – Illustration of the R^{2}CNN two-stage detector RoI pooling strategy.



Source: Jiang et al. (2017).

One-stage methods aim to simultaneously regress an OBB related to an object and predict its class label and are most popular than two-stage methods nowadays. Compared to HBBs, the regression step involves one additional parameter (angle), which impacts the design of the regression loss. For instance, Yang et al. (2021) proposed R3Det, which focuses on refining OBBs through pixel-wise feature interpolation. Qian et al. (2019) proposed the RSDet model, which uses the RetinaNet (LIN et al., 2017b) architecture with a new loss combined with an eight-parameter regression method (instead of the usual five-parameter) in order to solve the problem of inconsistent parameter regression in OBBs (e.g., the adoption of the angle parameter and the resulting height-width exchange, and the regression inconsistency of measure units in five-parameters models). Furthermore, in

order to solve the discontinuous boundaries issue (originated by the angular periodicity or corner ordering), Yang et al. proposed the Circular Smooth Label model (CSL) (YANG; YAN; HE, 2020) and the Densely Coded Labels model (DCL) (YANG et al., 2021) heads, which transform the angular prediction task from a regression to a classification problem.

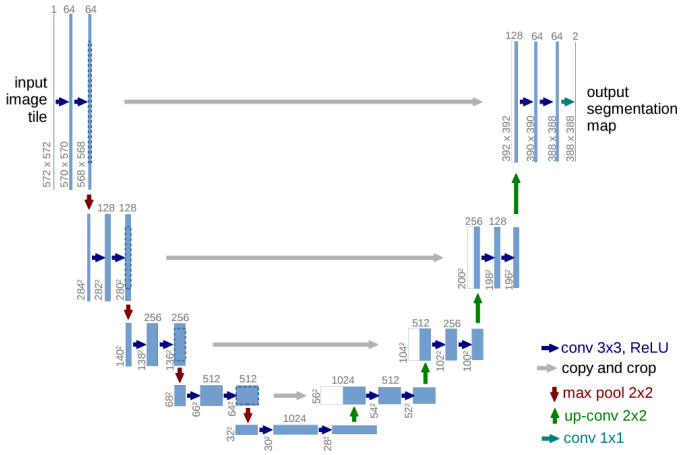
2.2 Cell detection and segmentation

There are several approaches for cell detection and segmentation, and the best results have been achieved by deep learning methods (ANOSHINA; SOROKIN, 2022; KÖRBER, 2022). These methods vary considerably regarding the underlying structure of the network and also on the degree of supervision required to label training data. For example, detecting just the cell nucleus requires one pixel-per-cell as supervision; detecting the cell boundaries as an HBB requires two points (e.g., top-left and bottom-right), while OBBs require an additional parameter related to the orientation; finally, segmentation requires the identification of all pixels belonging to a cell, which is very time-consuming. There are also some intermediate shape representations, such as the *star-convex polygons* (SCHMIDT et al., 2018), with intermediate annotation complexity.

The U-Net presented in (RONNEBERGER; FISCHER; BROX, 2015) has become a popular segmentation approach in biomedical applications when segmentation-level information is required. It is based on an encoder-decoder U-shaped architecture with skip connections (as shown in Figure 2.3), and focuses mostly on *semantic segmentation* tasks (i.e., segmenting all objects for each class altogether). Although the connected components produced by U-Net can also be used for *instance segmentation*, dense scenarios or situations with strongly overlapping cells are challenging. For instance segmentation, most approaches produce an embedding vector for each image pixel in such a way that similar vectors should relate to the same instance (object) (NEWELL; HUANG; DENG, 2017), and have been explored in the context of cell segmentation by Payer and colleagues (PAYER et al., 2018; PAYER et al., 2019) by using a cosine loss for estimating local embedding distances. To mitigate the cost of clustering pixels embedding (required to obtain the final instance segmentation), Zhao et al. (2021) proposed a fast Mean-Shift algorithm that works on GPUs. The *panoptic segmentation* task combines both category- and instance-level information into a single framework, and has been applied to biomedical imaging by Liu et al. (2021).

Although solutions that can return the full segmentation mask of cells have the clear

Figure 2.3 – U-Net architecture.



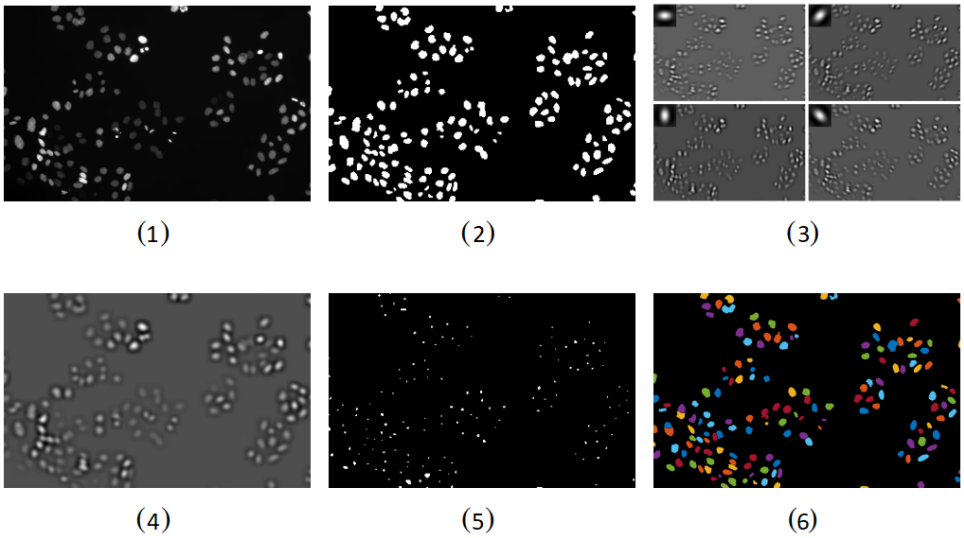
Source: Ronneberger, Fischer and Brox (2015).

advantage of providing a complete shape representation, most segmentation approaches are fully supervised, requiring time-consuming per-cell annotations. In order to overcome this problem, weakly supervised approaches explore partial annotations such as 3D HBBs (ZHAO et al., 2018), center-points (ZHAO; YIN, 2020) or user-defined scribbles (OH; LEE; JEONG, 2022) for the segmentation task. Nevertheless, many applications (e.g., tissue engineering (LU et al., 2021), and tracing the cell lineage trees) do not require the full description of the cell shape, but rather a position, size, and orientation descriptor are enough. This alleviates the annotation process, making it faster, less tedious, and more scalable.

As noted by Schmidt et al. (2018), a popular approach for cell detection in microscopy images is to use general-purpose object detectors based on HBBs, such as the works of Liu et al. (2016) or Redmon et al. (2016). Despite the constant evolution of object detectors (CARION et al., 2020; TAN; PANG; LE, 2020; WANG; BOCHKOVSKIY; LIAO, 2021; YANG; LI; GAO, 2022), the representation of cells as HBBs presents limitations in denser scenarios, particularly when oriented and elongated cells are present, as aforementioned. Intermediate representations between HBBs and full segmentation masks have also been explored for cell detection. For example, Akram et al. (2016) proposed a cell detection method based on fitting multiple elliptical filter banks, as illustrated in Figure 2.4. Schmidt et al. (2018) presented a polygonal shape representation based on radial sweeps with equidistant angles. A similar approach using splines instead of polygonal representations was presented by Mandal and Uhlmann (2021), obtaining smoother cell boundaries.

In this work, we advocate using EBB representations for cell detection, which

Figure 2.4 – Overview of the cell detection method proposed by Akram et al. (2016).



Source: Akram et al. (2016).

is a natural extension of the OBB representation but usually provides a better fit to the cell shape. In particular, roughly circular shapes induce a naturally ambiguous angular representation when OBBs are used since any rotated square fits equally to a circle (recall the cell on the left in Figure 1.1). For these cases, the EBB would reduce to a circle, mitigating the angular problem. Finally, it is worth mentioning that the output of any OBB detector can be mapped to an EBB, and OBB detectors have demonstrated exciting results in the past years (JIANG et al., 2017; XIA et al., 2018; YANG et al., 2021; YANG et al., 2021b). An OBB annotation is almost as simple as an HBB, with a considerable gain in shape representation, particularly regarding applications that involve elongated cells.

2.3 Cell tracking

For general-purpose tracking systems, the main challenges are related to vanishing objects, occlusions, and variations in motion and appearances (PARK et al., 2021). In cell tracking, further challenges are added, since the cells can suffer from apoptosis (cell death) and/or mitoses (cell division). Moreover, as observed by Li et al. (2008), the cell movements frequently vary with time, and standard methods for inferring the object movement (such as the Kalman filter) cannot be directly applied. Meanwhile, scenarios with a high density of cells are also common, which usually imply in cells that overlap and occlude each other. Due to this broad spectrum of challenges, cell tracking solutions might require a tailored solution for each cell type and application. As noted by Ulman et

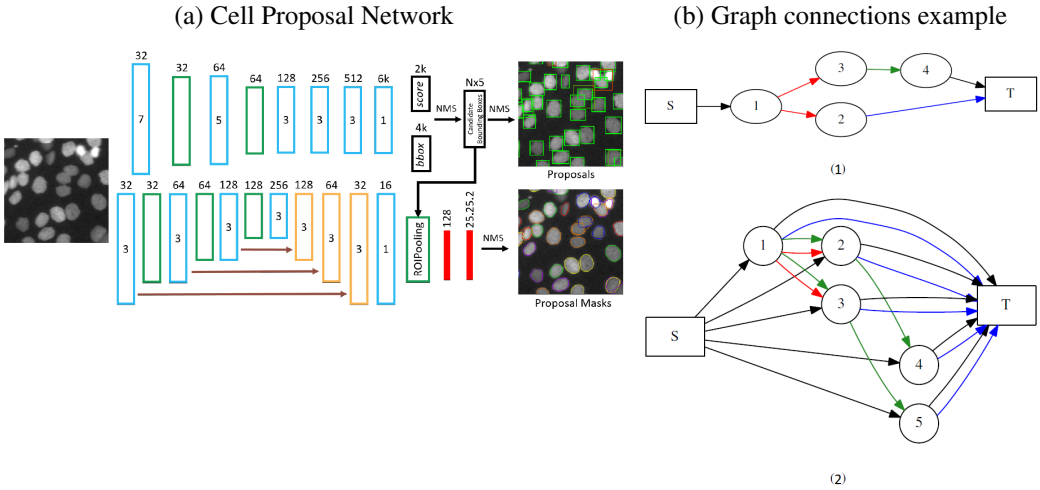
al. (2017), “there is no simple way to point out the right algorithm for a given dataset”, hinting that finding an algorithm capable of working for a broad spectrum of cell lineages is challenging. Nevertheless, the Cell Tracking Challenge (CTC) (MAŠKA et al., 2014) has provided several different datasets for benchmarking cell tracking algorithms, which usually are elaborated to work on more than one cell type and lineage.

Recent cell tracking algorithms can be broadly divided into two categories (WANG et al., 2020a): (i) tracking by model evolution and (ii) tracking-by-detection. In tracking by model evolution methods (a.k.a. end-to-end tracking systems), detection and tracking are solved simultaneously. In this context, Payer et al. (2018, 2019) introduced temporal information for spatio-temporal learning of the embeddings. They explored a cosine loss for estimating local embedding distances, and used a convolutional Gated Recursive Unit (ConvGRU) to learn temporal relationships. Nishimura et al. (2020) presented a cell tracking approach that works with weak annotations (cell centers in successive frames) by exploring a co-detection Convolutional Neural Networks (CNN). More recently, Hayashida, Nishimura and Bise (2022) proposed a complete pipeline that uses spatial-temporal context in multiple frames and long-term motion estimation with an objective level warping loss that addresses the problem of detecting and tracking highly dense cell images. Although these methods obtain most of the SOTA results, it is important to emphasize that they all require full annotations for both the detection/segmentation and tracking steps, which might be a strong limitation (e.g., requiring powerful hardware setups and long hours of training the models).

The tracking-by-detection paradigm consists of two stages: cell detection and cell association. When segmentation masks are required, the detection stage can either aid a segmentation step to extract the masks of each detected cell (AKRAM et al., 2017), or it can directly infer the segmentation masks and then split those wrongly joined cells using some algorithm such as the watershed (MAGNUSSON; JALDÉN, 2012; RONNEBERGER; FISCHER; BROX, 2015; BENSCH; RONNEBERGER, 2015; TÜRETKEN et al., 2015; AKRAM et al., 2016; BOUKARI; MAKROGIANNIS, 2018). The methods for connecting the cells in subsequent frames usually rely on graphs and integer linear programming (ILP) (MAGNUSSON; JALDÉN, 2012; TÜRETKEN et al., 2015; AKRAM et al., 2016; AKRAM et al., 2017; BOUKARI; MAKROGIANNIS, 2018) by defining the costs of the cell events (e.g., movement, mitoses, and apoptosis). Other strategies include using multi-Bernoulli random finite sets (XU et al., 2019) or joint particle filters based on Markov random field to model the dependency of the target movements (HIROSE et al., 2017).

Akram et al. (2017) proposed the Cell Proposal Network (CPN) method that uses a deep learning model that first detects the cells using the HBB representation and then feeds these detections to a segmentation model to further retrieve the segmentation masks of individual cells. For associating the detections, they use random forests to estimate the costs of the event graph, as illustrated in Figure 2.5. Boukari and Makrogiannis (2018) propose to detect the cells using a join spatio-temporal diffusion and region-based level-set optimization approach (BOUKARI; MAKROGIANNIS, 2016), and then track the cells using motion prediction and minimization of a global probabilistic function, as shown in Figure 2.6. More recently, Wang et al. (2020a) proposed a method that first segments the images using the U-Net (RONNEBERGER; FISCHER; BROX, 2015) model, and then uses deep reinforcement learning to associate the detected targets between frames.

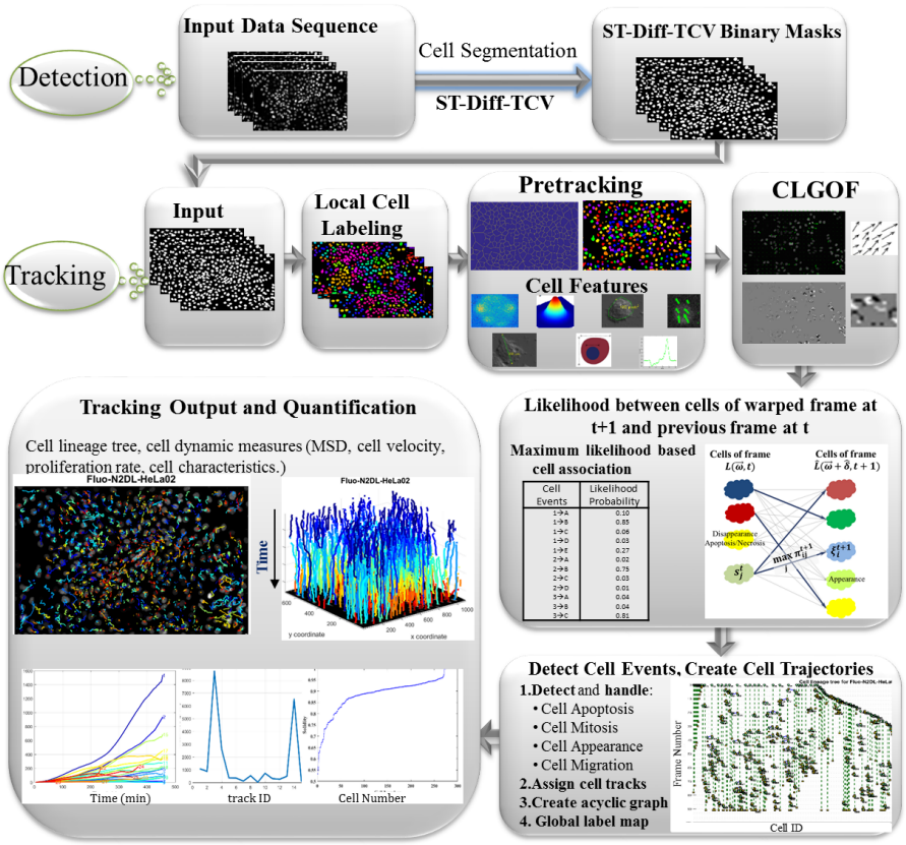
Figure 2.5 – Overview of the Akram et al. (2017) tracking-by-detection method. (a) Cell Proposal Network: first an HBB detector is used to propose cell regions to a second network segment the individual cell pixels in each detected region. (b) Graph connections example: (1) Ground Truth graph showing 4 cells and different possible events (represented by the different colors); (2) A proposal graph constructed from 5 proposals.



Source: Akram et al. (2017).

The current work proposes a tracking-by-detection approach that solely uses spatial information and the detection scores from an object detector to determine the associations between cells. Compared to methods that require pixel-wise segmentation masks (AKRAM et al., 2017; PAYER et al., 2019; NISHIMURA et al., 2020; WANG et al., 2020a; HAYASHIDA; NISHIMURA; BISE, 2022), it alleviates the annotation requirements to use only annotated cells as 5-parameter OBBs. Moreover, it also eliminates the necessity of using other features from the detections (e.g., image histograms) to compute the associations (TÜRETKEN et al., 2015; AKRAM et al., 2016; AKRAM et al., 2017), or to directly learn those features from the frames itself (PAYER et al., 2019; NISHIMURA

Figure 2.6 – Overview of the Boukari and Makrogiannis (2018) tracking-by-detection method. CLGOF stands for combined local/global optical optical flow, used to characterize and quantify the motion of objects.



Source: Boukari and Makrogiannis (2018).

et al., 2020; HAYASHIDA; NISHIMURA; BISE, 2022), which usually imply in more training time, bigger model size and robust hardware requirements. Our tracking system is based on the work of Bise, Yin and Kanade (2011), which uses a global data optimization algorithm to obtain the cell trajectories and lineage trees.

2.4 Evaluation metrics

We proceed to describe the standard metrics used in the current literature regarding object detection and tracking, which are used to evaluate our method and competitive approaches. We employed the following literature metrics for object detection: Precision (P), Recall (R), F1-Score (F1), and Average Precision (AP). The precision is defined as:

$$P = \frac{TP}{TP + FP}, \tag{2.1}$$

where TP is the number of True Positive detections and FP is the number of False Positives. Similarly, the Recall is defined as:

$$R = \frac{TP}{TP + FN}, \tag{2.2}$$

where FN is the number of False Negatives. The F1-score is the harmonic mean of these two metrics:

$$F1 = \frac{2 \times P \times R}{P + R}. \tag{2.3}$$

A TP or FP detection is defined by computing the overlap degree between two objects (i.e., predicted and ground-truth), and the Intersection over Union (IoU) is the usual choice. The IoU is defined as:

$$IoU(A, B) = \frac{A \cap B}{A \cup B}. \tag{2.4}$$

Following the standard protocol for general-purpose object detection, a predicted object is considered to be a TP if its IoU with one ground-truth object is above a certain threshold and the predicted category is correct (i.e., the predicted and ground truth objects are *matched*). Similarly, a FP detection corresponds to a predicted object that does not satisfy the previous conditions, and a FN corresponds to a ground-truth object with no matching predicted object. In this matter, it is usual to add the chosen IoU threshold as the subscript of the metric name. For example, P_{50} refers to computing the Precision using an IoU threshold of 0.5, while $P_{50:95}$ refers to averaging the computed Precision value using IoU thresholds from 0.5 to 0.95 with a 0.05 step.

However, note that the formulation of such metrics does not consider the confidence threshold of the predicted detection. For this reason, object detectors are usually evaluated using the AP metric, which is defined as the area under the precision-recall curve with varied score thresholds:

$$AP = \int_0^1 p(r)\mathrm{d}r, \tag{2.5}$$

where p is the precision-recall curve, and r is the score threshold. In order to avoid multiple evaluations of the Precision and Recall values on different score thresholds, this definition is usually simplified to an 11-point interpolated AP value, defined as:

$$AP \approx \frac{1}{11} \sum_{r \in S} AP(r) \tag{2.6}$$

where $S = \{0, 0.1, 0.2, ..., 0.9, 1.0\}$ are the score thresholds. Nevertheless, note that all of these metrics are closely related to the IoU threshold value used to match a predicted object to a ground-truth one. For this reason, they are usually computed using different values of IoU.

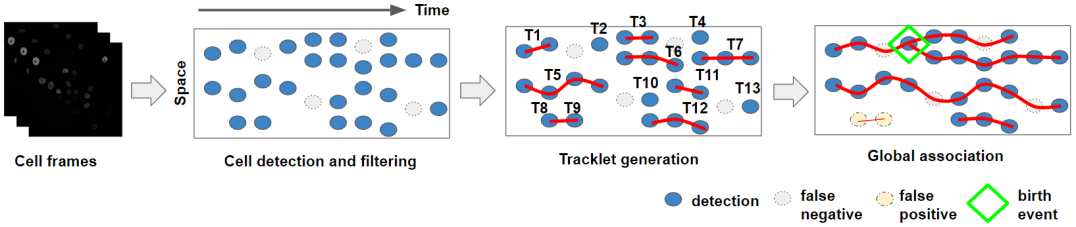
For the tracking evaluation, we used the CLEAR-MOT (MOTA) (BERNARDIN; STIEFELHAGEN, 2008; MILAN et al., 2016) and ID-MEASURE (the *MEASURE* relates to using the precision (IDP), recall (IDR) or F1-score (IDF1) metrics to its computation) (RISTANI et al., 2016) metrics*. Both metrics attempt to find a minimum-cost assignment between ground truth objects and predictions. However, while CLEAR-MOT solves the assignment problem on a local per-frame basis, ID-MEASURE solves the bipartite graph matching by finding the minimum cost of objects and predictions over all frames.

*Pyton implementation available at: <https://github.com/cheind/py-motmetrics>

3 THE PROPOSED TRACKING-BY-DETECTION METHOD

Our approach follows the typical pipeline of a tracking-by-detection method. First, an object detector is used to identify the cells in each frame using OBB representations, which are then converted to elliptical representations (EBBs). For tracking, we initially generate short tracklets by joining cells in subsequent frames with an objective function that jointly explores the shape and distance of cells. More precisely, we map the EBB representation to a two-dimensional Gaussian distribution and explore the Helinger distance, which directly correlates to the IoU metric (LLERENA et al., 2021). Finally, a global data association method based on the work of Bise, Yin and Kanade (2011) associates the tracklets to obtain the final cell trajectories and lineage trees. Figure 3.1 shows an overview of our complete pipeline for cell tracking-by-detection, and the steps are detailed next.

Figure 3.1 – Overview of the proposed pipeline for cell-tracking-by-detection. First, we detect the cells as OBBs and then convert them to the EBB representation. Next, we join the cells with high overlap between two adjecent frames. Finally, a global data association algorithm is used to identify all the cell events (i.e., movement, mitoses and apoptosis), while filling gaps generated by false negative detections and removing false positives ones, in order to produce the final tracklets.



Source: The authors.

3.1 Cell detection

The first step of our method relies on identifying the cells for each frame individually. We propose to use off-the-shelf OBB object detectors (see Section 4.4 for more details) trained with cell images and then convert the output to elliptical bounding boxes (EBBs). For an OBB with center (x, y), width W, height H, and orientation θ, we generate an ellipse with the same center and orientation, with semi-axes $a = W/2$ and $b = H/2$. If the OBB is clearly oriented (i.e., $W \gg H$ or $W \ll H$), the EBB will preserve the orientation of the OBB. On the other hand, if the OBB is roughly square (i.e., $H \approx W$), the produced EBB will be roughly circular. In the case of a perfect square, the EBB simplifies to a single circle regardless of the orientation of the OBB, which mitigates the orientation

ambiguity (recall the example shown in Figure 1.1).

3.2 Detection filtering and suppression

In a typical deep object detector, only candidate detections with scores larger than a pre-defined threshold τ_s are retrieved. Still, we usually have several overlapping candidates related to the same object, and Non-Maximum Suppression (NMS) is then used to retrieve only the candidate with the highest score. In this step, it is crucial to define a geometrical similarity measure between the detections for quantifying their "overlap" degree.

The Intersection-over-Union (IoU) is the *de facto* standard metric for computing the overlap in HBB or OBB detectors. However, computing the IoU for OBBs is not trivial due to the several possibilities for two intersecting OBBs (CHEN et al., 2020). Furthermore, the IoU is unreliable for OBB detections related to circular cells, since angular discrepancies might artificially degrade the IoU (MURRUGARRA-LLERENA; KIRSTEN; JUNG, 2022). Using the IoU with EBBs mitigates the latter problem, because the ellipse reduces to a circle. However, computing the intersection using EBBs is even more complex than using OBBs since it involves the overlap of two ellipses. In this work, we propose an alternative similarity metric based on fuzzy object representations.

In Yang et al. (2021a, 2021b) and Llerena et al. (2021), the core idea is to use 2D Gaussian distributions (denoted as GBBs – Gaussian Bounding Boxes) as intermediate representations for oriented objects, and train OBB detectors using loss functions based on similarity metrics between distributions. Here, we explore their developed fuzzy representation to compute the distance/similarity, as explained next. Following Llerena et al. (2021), an OBB with center $\boldsymbol{\mu} = (x_c, y_c)^T$, width W, height H and angle θ is mapped to a GBB described by the mean vector $\boldsymbol{\mu}$ (which is the OBB center) and a covariance matrix

$$\Sigma = \begin{bmatrix} a & c \\ c & b \end{bmatrix} = \begin{bmatrix} \frac{W^2}{12}\cos^2\theta + \frac{H^2}{12}\sin^2\theta & \frac{1}{2}\left(\frac{W^2}{12} - \frac{H^2}{12}\right)\sin 2\theta \\ \frac{1}{2}\left(\frac{W^2}{12} - \frac{H^2}{12}\right)\sin 2\theta & \frac{W^2}{12}\sin^2\theta + \frac{H^2}{12}\cos^2\theta \end{bmatrix}. \tag{3.1}$$

Note that square OBBs, for which $H = W$, generate a diagonal covariance matrix that does not involve the angular parameter θ. Hence, squares that differ only by angle are mapped to the exact same GBB.

In Yang et al. (2021a, 2021b) and Llerena et al. (2021), the Gaussian distributions are used to train OBB object detectors as their loss function. However, the metrics proposed by Yang et al. (2021a, 2021b) do not hold the mathematical properties of a

similarity measure, which is the goal for comparing two detection. Therefore, we define a similarity metric based on the Hellinger Distance, as done by Llerena et al. (2021). Let us consider that $p \sim \mathcal{N}(\boldsymbol{\mu}_1, \Sigma_1)$ and $q \sim \mathcal{N}(\boldsymbol{\mu}_2, \Sigma_2)$ are Gaussian distributions with

$$\boldsymbol{\mu}_1 = \begin{pmatrix} x_1 \\ y_1 \end{pmatrix}, \ \Sigma_1 = \begin{bmatrix} a_1 & c_1 \\ c_1 & b_1 \end{bmatrix}, \ \boldsymbol{\mu}_2 = \begin{pmatrix} x_2 \\ y_2 \end{pmatrix}, \ \Sigma_2 = \begin{bmatrix} a_2 & c_2 \\ c_2 & b_2 \end{bmatrix}. \tag{3.2}$$

The Bhattacharyya Distance (BHATTACHARYYA, 1946) between distributions p and q is given by

$$B_D(p,q) = \frac{1}{8}\boldsymbol{\mu}_{\mathbf{12}}^T \Sigma^{-1} \boldsymbol{\mu}_{\mathbf{12}} + \frac{1}{2}\ln\left(\frac{\det \Sigma_{12}}{\sqrt{\det \Sigma_1 \det \Sigma_2}}\right), \tag{3.3}$$
$$\boldsymbol{\mu}_{\mathbf{12}} = \boldsymbol{\mu}_1 - \boldsymbol{\mu}_2, \ \Sigma_{12} = \frac{1}{2}\left(\Sigma_1 + \Sigma_2\right),$$

and the Hellinger distance (HELLINGER, 1909) between p and q is then

$$H_D(p,q) = \sqrt{1 - e^{-B_D(p,q)}}. \tag{3.4}$$

Although both B_D and H_D are named "distances", only H_D satisfies the mathematical properties for a metric (KAILATH, 1967). Moreover, we can see that $0 \leq H_D(p,q) \leq 1$ (with 0 being the maximum similarity), meaning that H_D provides a normalized distance metric. In this work, we explore H_D to find overlapping EBBs to suppress non-maximum detections. Finally, given a detection p with the highest confidence score, we suppress all detections $q \neq p$ for which $H_D(p,q) < \tau_h$. We illustrate this procedure in Figure 3.2.

3.3 Tracklet generation

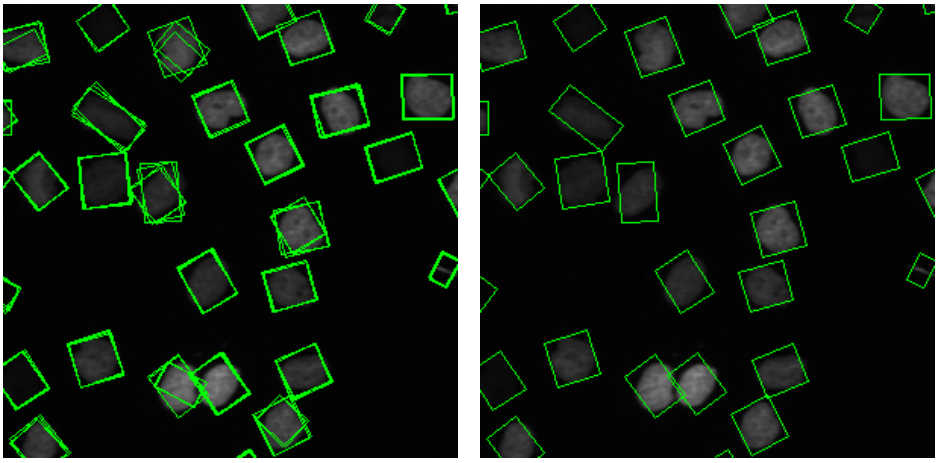
As stated by Bise, Yin and Kanade (2011), long trajectories obtained via frame-by-frame association may include more failures (such as drift and occlusions) than short trajectories. Hence, it is better to first reliably associate cell detections in adjacent frames, and then use some global data association algorithm to obtain the final long-term tracklets.

In this work, "reliable tracklets" are obtained by computing the overlap between all detections of subsequent frames with the Hellinger distance (Eq. (3.4)), and the optimal association between detections is obtained by solving a linear sum assignment problem with the Hungarian Algorithm (KUHN, 1955). Note that H_D jointly considers the centroid

Figure 3.2 – Illustration of detection filtering and NMS using the Hellinger Distance on a frame. Observe that in (a) we have many duplicates and false positives detection, while in (b) they are suppressed and eliminated, resulting in only one detection per cell.

(a) Frame with detections before filtering and suppression.

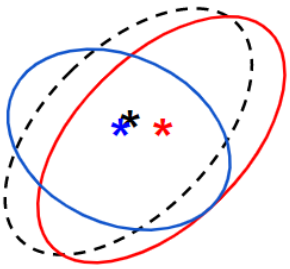
(b) Frame with detections after filtering and suppression.



Source: The authors.

distance (typically used for tracklet generation) and shape information – encoded in the covariance matrix. Hence, nearby cells with distinct shape/orientations that generate strong ambiguity when using only the centroid distance can be disambiguated through the Hellinger distance (see Figure 3.3).

Figure 3.3 – Illustration of using the Hellinger Distance for associating detections in subsequent frames. The dashed ellipse (black) represents a detection in frame i, while the two solid ellipses (red and blue) represent two possible associations in frame $i+1$ (the stars mark the respective centers of the detections). The Hellinger distance eliminates the imprecision of joining the black and blue detections due to only relating their centroid distances, while the red and black share more shape similarity.



Source: The authors.

In order to avoid bad associations caused by false positive detections, we only associate pairs of cells p and q for which $H_D(p, q) < \tau_o$, where τ_o is a similarity threshold. Note that we implicitly assume a low cell displacement and shape change in adjacent frames, which is typically the case for time-lapse microscopy imagery (BISE; YIN; KANADE, 2011).

3.4 Global data association

In an ideal scenario, the combination of the associations in adjacent frames would lead to the long-term track of each cell. However, there may be errors either in cell detection or the association process itself. Furthermore, we also have to consider cell division and death. Hence, we explore a global data association step to fill the gaps between disjointed tracklets, remove false positives and identify the mitoses events.

We based our method on the work by Bise, Yin and Kanade (2011), who formulated a maximum-a-posteriori problem (MAP) solved by linear programming that addresses the tree structure association problem. We proceed to briefly describe their method and then introduce our modifications. The MAP problem is solved by defining a set of hypotheses associated with a likelihood score for combinations of the N_T tracklets (i.e., each tracklet will respond to a set of possible hypotheses with their likelihood). More precisely, they assume the following five possible types of hypotheses that can be made for each tracklet: initiation, termination, translation, mitoses, and false positive.

Mathematically, let $\boldsymbol{C}_{M\times 2N_T}$ be a binary matrix containing the constraints for all possible hypotheses. Each row of $\boldsymbol{C}$ relates to a single hypothesis, and it presents $2N_T$ columns that indicate the possible tracklet associations, where the first N_T columns indicate the index of the source tracklet index and the following N_T columns relate to the target tracklet index (or indices). For example, the translation hypothesis presents one source tracklet and one target tracklet. Meanwhile, the mitosis hypothesis relates one source tracklet to a pair of target tracklets (its children), and the false positive hypothesis relates one tracklet to itself (i.e., the source index is the same as the target). There is also a likelihood value for each hypothesis (i.e., for each row of $\boldsymbol{C}$), stored in a vector $\boldsymbol{\rho}$ with M elements (the number of generated hypotheses).

The solution of the global optimization problem is a subset of all *non-conflicting hypotheses* (a subset of rows of $\boldsymbol{C}$) that maximizes the sum of the corresponding likelihoods. This can be formulated as the following ILP optimization problem:

$$\boldsymbol{x}^* = \underset{\boldsymbol{x}}{\operatorname{argmax}}\, \boldsymbol{\rho}^T\boldsymbol{x}, \quad s.t.\,, \boldsymbol{C}^T\boldsymbol{x} \leq \mathbf{1}, \tag{3.5}$$

where $\boldsymbol{x}_{M\times 1}$ is a binary vector, and an entry $x_k = 1$ means the k-th hypothesis is selected in the global optimal solution. The constraint $\boldsymbol{C}^T\boldsymbol{x} \leq \mathbf{1}$ guarantees that each tracklet ID appears in only one associated tree or false positive tracklet.

The work developed by Bise, Yin and Kanade (2011) allows a simple but efficient method for long-term data association. Yet, their approach still presents some limitations that we propose to address with the following modifications:

1. In Bise, Yin and Kanade (2011), they propose to use a specific algorithm (HUH et al., 2010) to identify the mitotic cells (i.e., the cells that are more likely to suffer mitoses). This algorithm, however, is specifically designed to work with images captured under phase-contrast microscopy, which narrows its usage to other types of images (e.g., fluorescence). We decided not to differentiate between mitotic and non-mitotic cells, and consider all cells as potentially mitotic. Furthermore, to address this choice, we employ two free parameters to adjust the likelihood distribution of the translation and mitoses hypothesis;
2. We removed the initiation and termination hypotheses and replaced them with a *completeness* one. In theory, these two hypotheses are essential to define the tree structure of the MAP problem. However, we noted in our experiments that they tend to generate a more complex MAP problem and do not provide better tree structures (i.e., closer to the ground-truth one), as we demonstrate in Section 5. In the original formulation, these hypotheses are defined regarding the cell position towards the borders and their first (or last) appearance, assuming that cells appear or disappear when they enter or leave the area captured by the microscope. Although these assumptions work well for an ideal detector, real detectors can fail to capture some cells (e.g., due to the absence of contrast between cell and background). As a consequence, the tracklets can start to be recognized only after some frames and/or with the cells already located farther away from the image boundaries, and imposing these boundary-related conditions can increase the number of false negatives;
3. We re-defined all the likelihood computations to use only information regarding the detected cells (e.g., position, time-frame distance, and confidence score returned by the object detector). In particular, we explore the confidence score to discriminate between the true and false positive hypotheses. For the translation and mitoses hypotheses, we used only the center and time distance between detections, whereas generic feature matching (e.g., based on histogram matching) was proposed by Bise, Yin and Kanade (2011). Our motivation is that finding an adequate feature to individually discriminate cells in long-term matching is challenging (no specific feature was mentioned in Bise, Yin and Kanade (2011)), and appearance-based features might change depending on the method used to capture the images. Moreover, using

only positional information and confidence score allow using any cell detector. We also removed the *true positive likelihood* that was originally proposed by Bise, Yin and Kanade (2011) since our pre-processing step can remove low-scored detections.

The proposed alternate hypotheses and corresponding likelihoods are explained next.

- **Translation hypothesis:**
  If the time and center distances between the last detection of tracklet X_{k1} and the first detection of X_{k2} are smaller than a pair of thresholds, $X_{k1} \rightarrow X_{k2}$ is a candidate of a tracklet translation. Considering that h denotes the index of a new hypothesis, we append a new row to $\boldsymbol{C}$ and a corresponding likelihood to $\boldsymbol{\rho}$ as:

$$C(h,i) = \begin{cases} 1, & \text{if } i = k_1 \text{ or } i = N_T + k_2 \\ 0, & \text{otherwise} \end{cases},$$

$$\rho(h) = P_{link}(X_{k2}|X_{k1}).$$

- **Mitosis hypothesis:**
  If the time and center distances between the last detection of tracklets X_p and the first detection of X_{c1} and X_{c2} are smaller than a threshold for the detection center and time, $X_p \rightarrow \{X_{c1}, X_{c2}\}$ is a candidate of a tracklet mitosis. We define new entries for $\boldsymbol{C}$ and $\boldsymbol{\rho}$ as:

$$C(h,i) = \begin{cases} 1, & \text{if } i = p \text{ or } i = N_T + c_1 \text{ or } i = N_T + c_2 \\ 0, & \text{otherwise} \end{cases},$$

$$\rho(h) = P_{mit}(X_{c1}, X_{c2}|X_p).$$

- **False positive and Completeness hypothesis:**
  If the score of tracklet X_k (i.e., the mean score of all its detections) is smaller than threshold τ_{FP}, X_k is a candidate for both false positive and completeness. We define two new entries for $\boldsymbol{C}$ (h and $h+1$) and $\boldsymbol{\rho}$ given by

$$C(h,i) = C(h+1,i) = \begin{cases} 1, & \text{if } i = k \text{ or } i = N_T + k \\ 0, & \text{otherwise} \end{cases},$$

and for the $\boldsymbol{\rho}$ entries as:

$$\rho(h) = P_{FP}(X_k),$$
$$\rho(h+1) = P_{cplt}(X_k).$$

The completeness hypothesis aims to cover the cases in which a tracklet does not translate or suffer from mitoses, i.e., the tracklet is, in fact, a full long-term cell track. Hence, assuming only the false positive hypothesis would wrongly eliminate those tracklets.

Now, we formalize the likelihoods for each hypothesis provided above. For the link and mitosis hypotheses, we would like to maximize the likelihood (i.e., values closer to 1) when the time and space distance between two cell tracklets are small, and minimize it (i.e., values closer to 0) when their distance increase. For this purpose, we based the formulation of the link and mitoses likelihoods on an exponential mapping that penalizes the time-space distance between tracklets. More precisely, we propose to use

$$P_{link}(X_j|X_i) = \exp\left(-\frac{(c_{i,j}+1)\, t_{j,i}}{\Delta t\ \lambda_{link}}\right), \tag{3.6}$$

where Δt is the dataset capture time step in $\frac{\text{frames}}{\text{hour}}$, $c_{i,j}$ is the Euclidean center distance between the last and the first detection of tracklets X_i and X_j, respectively, $t_{i,j}$ is their time distance in frames, and λ_{link} is a parameter that controls the decay of the exponential.

The mitosis likelihood is defined in a similar way, but jointly considering the space-time distance between the parent cell p and the two candidate children $c1$, $c2$. Formally, it is given by

$$P_{mit}(X_{c1}, X_{c2}|X_p) = \exp\left(-\frac{(c_{p,c1}+c_{p,c2}+1)(t_{p,c1}+t_{p,c2})}{4\Delta t\ \lambda_{mit}}\right), \tag{3.7}$$

where λ_{mit} controls the decay of the exponential. We do not use the Helinger distance in these formulations because cell shapes can change considerably in longer-term translations or during mitosis.

For the false positive likelihood, we also explore the precision α of the object detector computed for a given dataset (e.g., the training or validation set) and the tracklet score s_i, defined as the mean confidence score of all its detection. We want low-confidence tracklets (w.r.t. to the detector precision) to have an increased false positive likelihood, but longer tracklets must have a smaller value since they tend to be related to actual tracklets.

Based on these assumptions, the false positive likelihood is computed as

$$P_{FP}(X_i) = (1 - \alpha)(1 - s_i + \tau_s)^{|X_i|}, \tag{3.8}$$

where $|X_i|$ is the number of total detection responses in the tracklet, and τ_s is the threshold detection score defined in Section 3.2. Note that detectors with low precision ($\alpha \ll 1$) are prone to produce false positives, and P_{FP} should be increased in this case. On the other hand, detectors with high precision ($\alpha \approx 1$) are less prone to produce false positives, and the likelihood P_{FP} is decreased.

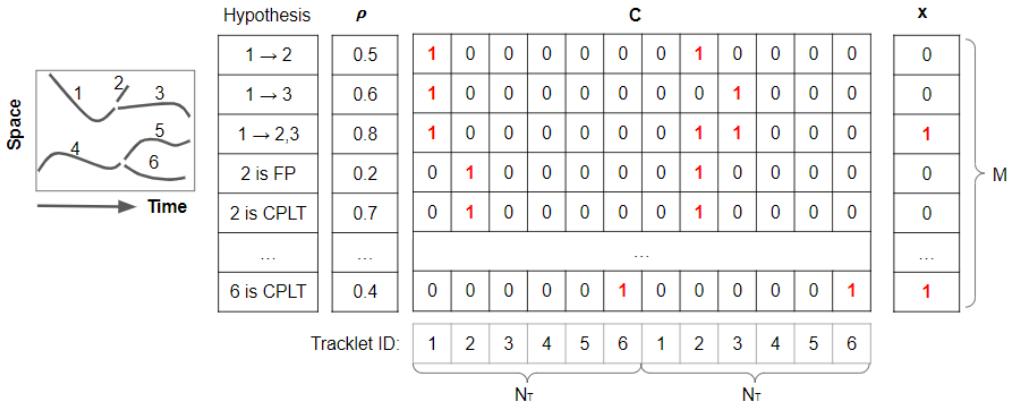
A similar rationale is used to define the completeness tracklet likelihood, given by

$$P_{cplt}(X_i) = \alpha(s_i - \tau_s). \tag{3.9}$$

Note that the likelihood of a tracklet being complete (i.e., it does not fit any other hypotheses) is not directly related to the tracklet size, since a cell can emerge and die very soon or appear in the visible field only in the last frames, which might lead to very small tracklets. Hence, we chose to define the likelihood based only on how precise the predictions of the object detector are, so that the tracklet generation step can do most of the correct associations between adjacent frames.

Figure 3.4 shows an example of the proposed hypotheses generation. From left to right, it shows a set of tracklets; the possible hypotheses connecting the tracklets; the likelihood vector $\boldsymbol{\rho}$; the matrix $\boldsymbol{C}$ indicating the possible tracklet connections; and the vector $\boldsymbol{x}$ returned by the MAP problem solution. The final tracks are then obtained by solving Eq. (3.5).

Figure 3.4 – Example of hypotheses generation for a given set of tracklets. Given a set of initial tracklets, we generate hypothesis for each of them, which are associated with a likelihood value stored in vector $\boldsymbol{\rho}$, and the constraints stored in the binary matrix $\boldsymbol{C}$. Solving the MAP problem returns the binary vector $\boldsymbol{x}$ which selects a subset of rows in $\boldsymbol{C}$ that defines the optimal solution of the MAP problem. For example, the first line defines a translation hypothesis of tracklet 1 to 2 with a likelihood of 0.5. The matrix $\boldsymbol{C}$ stores the ID values of the considered tracklets (in each of its halves), and the value 0 in the vector $\boldsymbol{x}$ indicates that this hypothesis was not chosen during the optimization solving.



Source: The authors.

4 EXPERIMENTS

4.1 Datasets

For evaluating OBB/EBB cell detectors, we used a private dataset (mostly composed of cancer cells) specifically annotated for the detection part of the pipeline, named *Oriented cell dataset* (OCD). For evaluating the full tracking-by-detection pipeline, we used three public datasets from the CTC (MAŠKA et al., 2014) challenge. In fact, they are used as a benchmark for detection, segmentation and tracking for cell applications. All these datasets are detailed next.

4.1.1 OCD private dataset

The OCD images were provided by the LabSinal laboratory* and annotated by students of biotechnology†. A total of 150 images were acquired using different microscopes and cell lineages, which resulted in visually distinct images as shown in Figure 4.1. These images were evenly split into training (120 images, 80% of total) and test (30 images, 20% of total) sets in order that each set contained an approximately equal distribution regarding the used microscope (CytoSMART and Zeiss Axiovert 200), lineage (human glioblastoma, human breast cancer, and human lung fibroblast), and cultivation method (cells cultivated without aiding chemotherapy, and cells treated with temozolomide in some cultivation step). A full description of the data acquisition and splits is described in Table 4.1.

The images were annotated using the roLabelimg tool‡, which allowed the biology researchers to delimit an OBB (composed of x-center, y-center, height, width and angle) over each cell. Furthermore, the cells were classified either as "normal" or "round" cells, for which the biological interpretation can vary depending on the lineage (usually, round cells are close to doing mitoses). Figure 4.2 illustrates this classification difference. In total, 6,461 OBBs were annotated, of which 5,810 (89,9%) were classified as "normal" and 651 (10,1%) as "round" cells. Since the annotations in this dataset provide only the OBBs for individual frames, it was only used in our preliminary investigations of using OBBs to detect the cells.

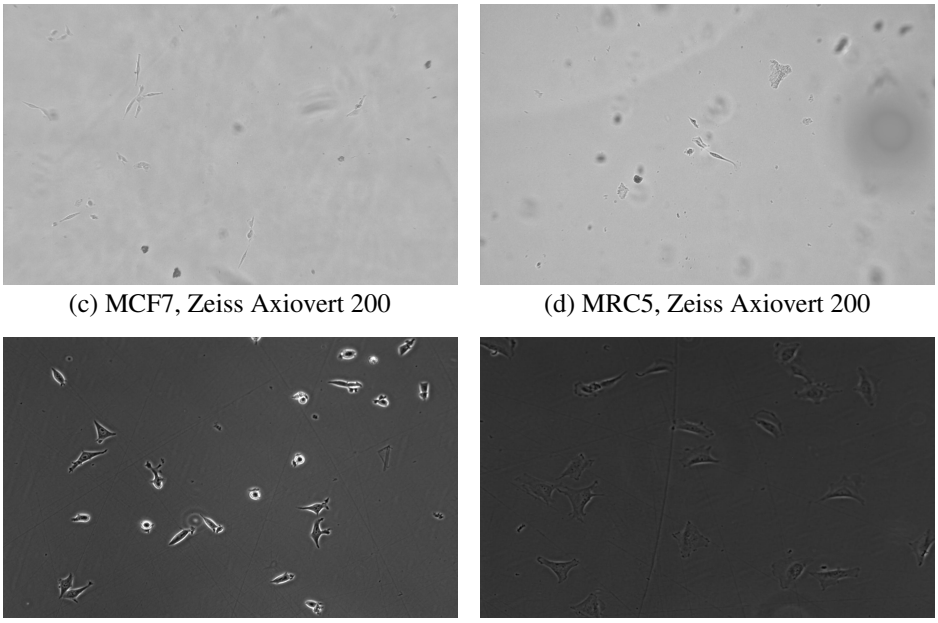
*Signaling and Cellular Plasticity Laboratory – UFRGS. Web-page: <https://www.ufrgs.br/labsinal/>

†Specifically, this dataset was generated by prof. Dr. Guido Lenz's research group, namely students Angelo Luiz Angonezi and Fernanda Dittrich Oliveira.

‡Available at <https://github.com/cgvict/roLabelImg>.

Figure 4.1 – Example of different images from the OCD, described as: lineage, microscope (see Table 4.1 for more details).

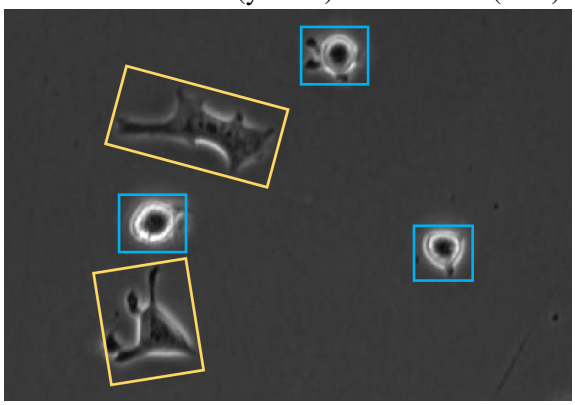
(a) A172, CytoSMART (b) U251, CytoSMART



(c) MCF7, Zeiss Axiovert 200 (d) MRC5, Zeiss Axiovert 200

Source: The authors.

Figure 4.2 – Illustration of “normal” (yellow) and “round” (blue) cells in the OCD.



Source: The authors.

4.1.2 CTC public dataset

For evaluating our complete method (i.e., detection and tracking), we used three publicly available cell microscopy datasets provided from the CTC (MAŠKA et al., 2014): Fluo-N2DH-GOWT1, PhC-C2DH-U373, and Fluo-N2DL-HeLa, illustrated in Figure 4.3. Each dataset contains two sequences in the training set (with public ground truth annotations) and two challenge sequences (with hidden ground truth annotation), named with suffices -01 and -02. The method performance for the challenge sequences are obtained by submitting the results to the CTC server. We proceed to provide details regarding each of the chosen datasets.

Table 4.1 – OCD description. Lineages A172 and U251: human glioblastoma; MCF7: human breast cancer; MRC5: human lung fibroblast. Cultivation condition CTR: cells cultivated without aiding chemotherapy; TMZ: cells treated with temozolomide in some cultivation step.

Split	Microscope	Zoom	Lineage	Cultivation Condition	Number of images	Images Resolution
Train	CytoSMART	20x	A172	CTR	52	1280×720px
				TMZ	25	
			U251	CTR	4	
	Zeiss Axiovert 200	10x	MCF7	CTR	19	1388×1040px
			MRC5	CTR	20	
Test	CytoSMART	20x	A172	CTR	4	1280×720px
				TMZ	5	
			U251	CTR	1	
	Zeiss Axiovert 200	10x	MCF7	CTR	10	1388×1040px
			MRC5	CTR	10	

Source: The authors.

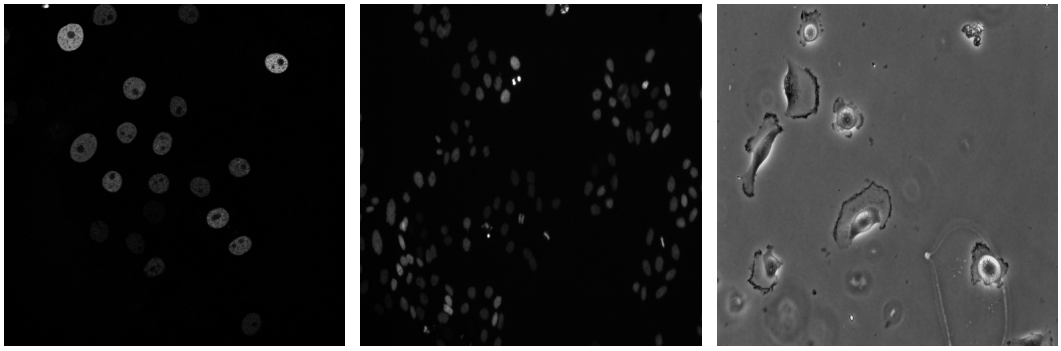
- Fluo-N2DH-**GOWT1** contains GFP-transfected GOWT1 mouse embryonic stem cells captured on fluorescence microscopy. Challenges with this dataset include low contrast of some cells and few cells entering and exiting the imaged region from the axial direction. The capture time step is $\Delta t = 12\frac{\text{frames}}{\text{hour}}$, and the images have resolution 1024×1024px. There are 92 images on both sequences.
- PhC-C2DH-**U373** contains glioblastoma-astrocytoma U373 cells captured under phase contrast microscopy. This dataset is challenging due to cells having highly deformable shapes and parts of cell bodies having a similar appearance to the background. The capture time step is $\Delta t = 4\frac{\text{frames}}{\text{hour}}$, and the images have resolution 696×520px. There are 115 images on both sequences.
- Fluo-N2DH-**HeLa** contains fluorescently labeled HeLa nuclei captured on fluorescence microscopy. Challenges with this dataset include high cell density, low contrast, a few irregularly shaped cells, various mitoses events, and cells entering and exiting the imaged region. The capture time step is $\Delta t = 2\frac{\text{frames}}{\text{hour}}$, and the images have resolution 1100×700px. There are 92 images on both sequences.

The time step for all these datasets is reasonable with our assumptions of low cell movement in subsequent frames.

All these datasets only contain ground truth (GT) annotations for cells within a field of interest, which excludes a few pixels for cells close to the image boundaries. There are two types of GT annotations: cell *masks* for the segmentation evaluation, and cell *markers* for detection and tracking evaluation. For the cell masks, the annotations are provided as *silver* and *gold* standards. The silver standard annotations refer to computer-originated

Figure 4.3 – Example of cell images for each of the used datasets from the CTC (MAŠKA et al., 2014).

(a) Fluo-N2DH-**GOWT1** (b) Fluo-N2DH-**HeLa** (c) PhC-C2DH-**U373**

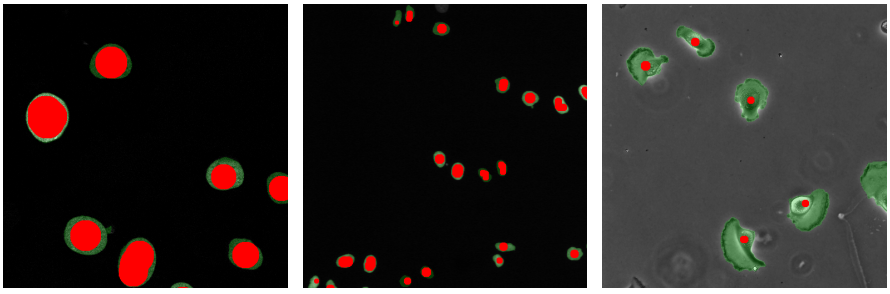


Source: Maška et al. (2014).

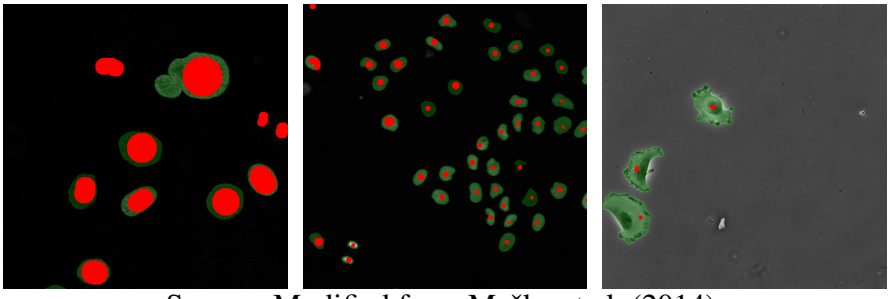
reference annotations, while the gold standard refers to human-originated ones. Since only a few cells are annotated in the gold standard, we used only the silver ones for both training and evaluation, as in previous works (MAGNUSSON; JALDÉN, 2012; TÜRETKEN et al., 2015; AKRAM et al., 2016; AKRAM et al., 2017; WANG et al., 2020a). The cell markers are "similar" to the segmentation masks, but they have reduced size and serve solely as a positional descriptor of the cells. Moreover, they do not follow a standard regarding their size or placement on the cell image, which might impact the quality metrics. We illustrate these annotation discrepancies in Figure 4.4. Note that the markers sometimes are very close to the full segmentation masks, but relate to small regions in others.

Figure 4.4 – Illustration showing the difference between the two CTC (MAŠKA et al., 2014) annotations types: cell masks (green) and cell markers (red).

(a) GOWT1-01 (b) HeLa-01 (c) U373-01



(d) GOWT1-02 (e) HeLa-02 (f) U373-02



Source: Modified from Maška et al. (2014).

4.2 Data pre-processing

We use the same data pre-processing procedure for all datasets, except for some specific adjustments in the HeLa, as described next. Since the HeLa dataset contains cells with very different sizes compared to the other CTC datasets (MAŠKA et al., 2014), we followed a similar strategy to the one employed by Akram et al. (2017) and magnified all images by a factor of 2. This procedure allows us to use the same network architecture without needing any adjustment in its head parameters regressor (e.g., the anchors) to better fit the cell sizes in this dataset.

For training the object detector, we extracted patches of full images in the datasets using a 512×512px sliding window with 100px stride. The patches are extracted starting the window at the top-left corner of the image and sliding it across the image horizontally and vertically according to the specified stride. At each position, we extract a patch within the window boundaries. This process resulted in $5{,}590$ training images for the OCD and, for each sequence of the CTC dataset, $4{,}508$ for the GOWT1 dataset, 690 for U373, and $16{,}314$ for HeLa. During inference, we used the full-sized images, except for the HeLa dataset, for which we used the patches with a 256 stride (to provide some overlap in the borders) and then divided the parameters of the detection by a factor of 2 to retrieve them with the expected original image sizes. For the CTC datasets, we used a similar strategy as in other works (MAGNUSSON; JALDÉN, 2012; TÜRETKEN et al., 2015; AKRAM et al., 2016; AKRAM et al., 2017; WANG et al., 2020a): with dataset sequences that contain GT annotations (i.e., not the one related to the challenge), we trained the models with one sequence and used the other for evaluation. For the OCD, we already had established training and testing sets. To generate the OBBs (and then the EBBs) from the CTC cell masks, which are required to train the object detectors, we employed the minimum-area rectangle fitting algorithm available in the OpenCV (BRADSKI, 2000) framework.

4.3 Evaluation protocol

We evaluated our results using standard literature metrics employed for evaluating general-purpose detection and tracking systems (as described in Section 2.4), and the metrics proposed by the CTC (MAŠKA et al., 2014). The CTC provides the DET, SEG,

and TRA metrics[§]. Both DET and TRA metrics are designed to mirror the manual effort required to correct the errors of a given detection and tracking algorithm, respectively, using Acyclic Oriented Graph Matching; SEG measures the Jaccard similarity index (a.k.a. IoU) between predicted and ground-truth segmentation masks. All CTC metrics return values from 0 to 1 (1 being the highest score).

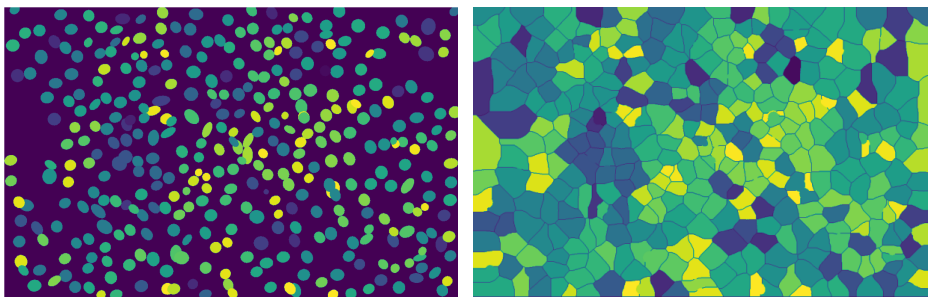
Since our method provides only an approximation of the segmentation masks through EBBs, we would also like to estimate how close both the GT EBBs and the predicted EBBs are to the GT segmentation masks. The former can be answered by evaluating the EBBs generated from the ground truth cell masks with the SEG metric, obtaining the *EBB SEG* metric. The latter is obtained by simply evaluating our method with the SEG metric.

For the detection and tracking evaluation, the algorithm provided by the CTC disregards detections that do not entirely overlap with the provided ground-truth cell markers. As mentioned before, they do not follow a standard for size and displacement, which might affect the quantitative metrics. For the U373-02 dataset, in particular, Akram et al. (2017) propose simply enlarging the predicted masks to avoid missing the cell markers. In our approach, however, the variability of cell mark annotations can significantly impact all the tested datasets, since the EBBs are only approximations of the segmentation mask and might not completely overlap with a ground-truth mark – they are not bounding representations as well. In order to overcome this issue, we also evaluated our method that enlarged the predicted cell masks with a simple watershed algorithm, using the EBBs as guiding markers. An example of EBB-guided watershed is shown in Figure 4.5.

Figure 4.5 – Visual comparison of the cell masks using EBBs (left) and post-processed with watershed (right). Note that the watershed fills the voids between the different detections allowing some degree of error when evaluating with the DET and TRA metrics.

(a) Cell masks using EBBs.

(b) Cells masks post-processed with watershed



Source: The authors.

[§]A full description of the metrics can be found at: <http://celltrackingchallenge.net/evaluation-methodology/>

4.4 The tested OBB detectors

The proposed tracking-by-detection scheme can use any OBB detector. In this work, we selected a total of seven general-purpose SOTA OBB detectors provided in the AlphaRotate rotation benchmark[¶], and trained them using the OCD. This experiment allowed us to identify which object detector could be more suitable for the proposed tracking-by-detection pipeline applied to public datasets, since the OCD contains a high variability in terms of image appearance and in the variability of cell shapes as well. The chosen anchor-based detectors (in chronological order) are briefly described next:

- **R^{2}CNN** (JIANG et al., 2017) is a model proposed for detecting arbitrary-oriented texts in natural scene images. It is based on the Faster R-CNN (REN et al., 2016) architecture and includes blocks of pooling and feature concatenation by the end of the Region Proposal Network (RPN) module in order to estimate the OBBs using an *inclined non-maximum suppression* algorithm;
- **RetinaNet** (LIN et al., 2017b) is a single, unified network composed of a backbone model and two task-specific sub-modules: one for object classification, and the other for OBB regression. It also proposes the usage of the *Focal Loss* as a form to overcome foreground-background class imbalance encountered during the training of dense detectors. Originally, this model was developed to work only with HBB detection. To also work with OBB detection, an extra parameter (angular) is necessary to be regressed, as well as the aid of new anchors;
- **RSDet** (QIAN et al., 2019) uses the RetinaNet architecture with a new loss combined with an eight-parameter regression method (instead of the usual five-parameter) in order to solve the problem of inconsistent parameter regression in OBBs (e.g., the adoption of the angle parameter and the resulting height-width exchange, and the regression inconsistency of measure units in five-parameters models);
- **CSL** (YANG; YAN; HE, 2020) proposes to solve the discontinuous boundaries issue (originated by the angular periodicity or corner ordering) by transforming the angular prediction task from a regression to a classification problem. In order to achieve this, it uses a *Circular Smooth Label* technique (that uses a *Gray Coded Label* method), altogether with an appropriate new head module;
- **DCL** (YANG et al., 2021) similarly to CSL, it intends to solve the boundary dis-

[¶]Available at: <https://github.com/yangxue0827/RotationDetection>

continuity issue on OBB detectors through a new encoding mechanism for angle classification instead of regression, but also employing a loss re-weighting scheme (by proposing an *angle distance* and *Aspect Ratio Sensitive Weighting*). These changes were built on top of the RetinaNet architecture;

- **R3Det** (YANG et al., 2021) is an end-to-end refined single-stage rotation detector that uses a progressive regression approach from coarse to fine granularity. It implements a feature refinement module to improve detection performance and also proposes an *approximate SkewIoU loss* for regressing the OBBs;
- **R3Det DCL** (YANG et al., 2021; YANG et al., 2021) uses the same parametrization employed by the DCL model, but, instead of using RetinaNet as base architecture, it uses R3Det (YANG et al., 2021).

All models use the ResNet50 (HE et al., 2016) backbone with pre-trained weights on the ImageNet dataset (DENG et al., 2009). Weight decay and momentum are set to 10^{-4} and 0.9, respectively. Both classification modules (RPN and head) use the categorical cross-entropy loss, and both box regression modules use the smooth-ℓ_1 loss. We employ Momentum Optimizer over one GPU and one image per mini-batch. All models are trained for 20 epochs for $5{,}000$ steps. The learning rate started at 10^{-3} and reduced tenfold at 12 epochs and 16 epochs, respectively. Finally, we applied random rotation (sampled from a range of $[-90, 90]$ degrees, with 15 degree step), and flips (vertical and horizontal) to augment the training data in all experiments.

The results for testing different object detectors on the OCD are presented in Table 4.2. We can observe that most models returned close AP and recall values, but there is a huge gap between the precision and F1-score of the R^{2}CNN model compared to the other ones. More precisely, the precision of the R^{2}CNN is $1{,}7$ times higher and the F1-score is $1{,}4$ for detecting normal cells, while $3{,}2$ and $2{,}3$, respectively, times higher for detecting round cells than the second best model (DCL in all cases). Because of this huge difference on the precision and F1-score results, and the small differences in the other ones (recall and AP), we decided to use the R^{2}CNN model as the object detector for our tracking-by-detection system.

Table 4.2 – Results for the OCD using SOTA OBB detectors. **Bold** values mark the best results, while underline values mark the second best.

Model	normal				round				AP_{50}	AP_{75}	$AP_{50:95}$
	AP_{50}	P_{50}	R_{50}	$F1_{50}$	AP_{50}	P_{50}	R_{50}	$F1_{50}$			
DCL	71.81	39.58	81.62	53.31	63.35	16.50	80.95	27.41	67.58	17.74	28.48
CSL	72.42	35.20	84.10	49.63	53.98	10.94	84.76	19.38	63.20	18.59	27.00
RSDet	74.33	38.65	83.31	52.80	63.60	9.63	87.62	17.35	68.97	16.15	29.86
RetinaNet	75.25	37.64	85.00	52.18	56.47	10.99	82.85	19.41	65.86	19.17	29.56
R3Det	75.15	27.93	**85.68**	42.13	66.75	11.94	**89.52**	21.07	70.95	18.86	**31.00**
R3Det DCL	75.50	31.58	83.54	45.83	62.63	15.75	84.76	26.56	69.06	**20.57**	30.73
R^2CNN	**77.33**	**67.49**	82.41	**74.21**	**69.65**	**52.79**	80.95	**63.91**	**73.49**	17.30	29.71

Source: The authors.

4.5 Tracking-by-detection implementation details

As the object detector, we chose to use the R^2CNN (JIANG et al., 2017), because it has shown to have the best compromise among precision, recall and AP compared to other OBB detectors in our preliminary investigations using the OCD (see Table 4.2). Furthermore, as presented by Liu et al. (2020), general-purpose two-stage detectors usually provide better results when compared with one-stage ones. The model parameters used for all experiments are the same as the ones used for the OCD. All models are trained for 100 epochs with a 0.1 reduction factor of the learning rate at epochs 12, 16, and 20, using random rotation and flips as data augmentation primitives. The initial learning rate was 10^{-3} for all datasets except HeLa, for which the model showed overfitting. In this dataset, we used 10^{-4} as the initial learning rate and trained for 24 epochs only. For evaluating the object detector, we used the Hellinger distance (Eq. (3.4)) to compute the overlap between predicted and ground-truth detections, and set the score and overlap thresholds $\tau_s = \tau_h = 0.5$.

The hyper-parameters for the tracking system are the same for all datasets (except the object detection precision α and capture time Δt), and were chosen empirically to produce good results for all the evaluated datasets (i.e., we used the training datasets in order to define the hyper-parameters of the final tracking system that was evaluated on the CTC (MAŠKA et al., 2014) server with hidden GT). Nevertheless, we provide a sensitivity analysis of the parameters in Section 5.3, and conclude that changing them has a small impact on the results. For the inference of the detector, we used a score threshold $\tau_s = 0.5$, and an overlap threshold of $\tau_h = 0.5$ for the filtering and aggregation step. For the tracklet generation, we employed an overlap threshold of $\tau_o = 0.5$, which are classical thresholds for IoU-like metrics.

For the parameters of the global data association algorithm, we used a time thresh-

old $t_{th} = 3$ frames, and a false positive threshold $\tau_{FP} = 0.9$. The space threshold was set to $0.1\sqrt{W_f^2 + H_f^2}$, where W_f and H_f are the width and height of the dataset frames, respectively. The α value was computed using the Eq. (2.1) for each dataset considering an overlap threshold of $\tau_h = 0.5$ between the predicted and ground-truth detections of the training images with no score threshold (i.e., considering any detection with a confidence score above zero), and are available in Table 4.3. The free parameters for likelihood adjustment are set to $\lambda_{link} = 25$ and $\lambda_{mit} = 50$ for all datasets. The MAP problem was solved using the Cbc (FORREST et al., 2022) mixed ILP solver provided in the CVXPY[‖] Python 3 (ROSSUM; JR, 1995) library.

Table 4.3 – Object detection precision α for each dataset.

Dataset	GOWT1-01	GOWT1-02	U373-01	U373-02	HeLa-01	HeLa-02
$\boldsymbol{\alpha}$	0.8993	0.8644	0.7673	0.6867	0.7930	0.7899

Source: The authors.

[‖]Available at: <https://www.cvxpy.org/index.html>

5 RESULTS AND DISCUSSIONS

In this section, we present the results for evaluating our tracking-by-detection method in the CTC (MAŠKA et al., 2014) datasets. We evaluated our results in two manners: one by using only the training datasets with GT and comparing with other works directly (see Section 4.2 for more details), and the other by submitting our codes to the CTC server* in order to retrieve the metrics scores and rank compared to other submitted works. We begin this section by briefly describing the baseline methods used in the first evaluation methodology. Then, we present the results for both methodologies and discuss them.

5.1 Baseline methods

We proceed to briefly describe the methods used as a baseline for comparison with our proposal tracking-by-detection method on the CTC datasets (MAŠKA et al., 2014), highlighting the nature of required training data.

KTH (MAGNUSSON; JALDÉN, 2012) segments cells using a bandpass filter followed by thresholding, and then uses the watershed algorithm to split joined cells. The tracking graph is created by connecting cell segmentations in adjacent frames, and then solved by iteratively finding the lowest cost path in the graph using Viterbi algorithm. It does not require any annotation.

EPFL (TÜRETKEN et al., 2015) detects cells by fitting ellipses to binary segmented regions. It joins the detection on subsequent frames using a tracking graph solved using ILP. It requires annotation for segmentation, detection, and tracking.

HEID (TÜRETKEN et al., 2015) detects cells by merging super-pixels clustering segmentation, which are obtained using watersheds. Then, the tracking is retrieved by finding the global optimum of a graph model that represents cellular events using ILP. It requires annotation for segmentation, detection, and tracking.

BLOB (AKRAM et al., 2016) detect cells using multiple elliptical filter banks, and performs tracking by iteratively finding the shortest path in a model graph. It requires only tracking annotation, i.e., temporal cell associations.

CPN (AKRAM et al., 2017) first generates cell region proposals using an HBB object detector, and then finds the segmentation masks of these regions using a deep

*Available at: <http://celltrackingchallenge.net/>

learning segmentation network similar to the U-Net (RONNEBERGER; FISCHER; BROX, 2015). It performs tracking using ILP to solve a graph for which the weights are set by training a random forest classifier with several histogram features. It requires full annotation for detection, segmentation, and tracking.

ST-TCV (BOUKARI; MAKROGIANNIS, 2018) detects cells using a joint spatio-temporal diffusion and region-based level-set optimization approach (BOUKARI; MAKROGIANNIS, 2016). Then, it uses motion prediction and minimization of a global probabilistic function to join the cells of subsequent frames. It does not require any annotation.

DRL (WANG et al., 2020a) uses the U-Net (RONNEBERGER; FISCHER; BROX, 2015) model to produce the cell segmentation masks. Then, it uses deep reinforcement learning to build the cost matrix that joins the cells of subsequent frames. It requires full annotation for detection, segmentation, and tracking.

We did not include results for the methods U-Net (RONNEBERGER; FISCHER; BROX, 2015), GC-ME (BENSCH; RONNEBERGER, 2015), and U-Net S (GUPTA et al., 2019) because they do not follow the same data split methodology (i.e., they employ images from both training sequences of each cell lineage in the training phase), and hence it would define an unfair baseline for comparison.

5.2 Results for the CTC datasets

The results for comparing our method with SOTA approaches on the CTC evaluation using separate sets are provided in Table 5.1. We include in this table results for both using or not using the watershed algorithm for enlarging the detection results. For each dataset, we provide the results regarding the evaluation on one sequence, while training with the other (e.g., GOWT1-01 means that we trained the object detector with sequence 02 and tested in 01).

Regarding the approximation of the cell masks using EBBs, we can observe from the *EBB SEG* column that it provides a good fit for the GOWT1 and HeLa datasets. On the other hand, the EBB approximation is not very good for the U373 dataset since there is strong variability in the cell shapes, as mentioned before. Finally, our method did not achieve SOTA scores on the SEG metric, which is expected since we only approximate the cell masks through EBBs. Nevertheless, it could reach close values to those on both the GOWT1 and HeLa datasets, and even get the second best for the GOWT1-01.

For the DET and TRA metrics, we note that our approach achieves a considerable

boost using the watershed post-processing algorithm, particularly for the U373-02 dataset. We believe that this behavior is mostly due to the GT annotation of the cell markers in the dataset, which is sometimes located at the boundary of the cells and might not overlap completely with the EBB. Nevertheless, our method (without watershed) could reach SOTA results in both GOWT1 datasets, while having the second-best result in U373-01. When applying the watershed mask augmentation, our method reached SOTA scores on three datasets, and second best on other two. Regarding the degree of annotation required for each technique, our method was capable of outperforming fully supervised methods (HEID (TÜRETKEN et al., 2015), EPFL (TÜRETKEN et al., 2015), CPN (AKRAM et al., 2017) and DRL (WANG et al., 2020a)) in most of the evaluated datasets. It also presented better results than the tracking-supervised approach BLOB (AKRAM et al., 2016) and the unsupervised trackers KTH (MAGNUSSON; JALDÉN, 2012) and ST-TCV (BOUKARI; MAKROGIANNIS, 2018) in all datasets, except for HeLa-02 compared to the KTH method.

Table 5.2 report the results for evaluating our method on the CTC server [†] on the DET and TRA metrics with and without the watershed method[‡]. In this evaluation, our method was capable of achieving the TOP 3 rank on the DET metric for the GOWT1 dataset using the watershed algorithm. Although it could not overcome the SOTA in any dataset, we can observe a small difference between the scores of our method with those ranked as the top one. Furthermore, most of the top-ranked algorithms are end-to-end trackers or use elaborated techniques to improve the predicted segmentation masks from deep learning models (e.g., using model assemble or multiple refinement stages). In contrast, our proposed method intends to provide a simple yet efficient method for tracking-by-detection that requires only per-frame OBB cell annotations.

The results using standard detection and quality metrics are provided in Table 5.3. The detection results refer only to the detector performance itself, i.e., it does not use the global data association to further eliminate false positive detections and/or add false negative ones. This table evaluates different aspects of our method, enabling us to identify its strengths and weakness better. Regarding the recall metric, we can observe that it could achieve high values on all datasets for detection and tracking. However, it is noticeable that our method fails to eliminate false positive detections, which impact the precision

[†]Available at: <http://celltrackingchallenge.net/participants/UFRGS-BR/>

[‡]Due to environment problems related to CUDA instructions on the CTC server computers, we could not reproduce the exact same code used on our side. This ended up slightly harming the predicted EBB shapes and hence under-estimating the SEG metric, and the DET and TRA metrics when the detections' shapes are not augmented with the watershed algorithm.

in both detection and tracking, as noted for dataset U373-01. On the other hand, the detector precision in the GOWT1-02 dataset was also relatively low, but our global data association algorithm was capable of eliminating most of the false positive detections and hence obtaining a higher precision value on the tracking metrics. For U373-02, we can observe an inconsistency between the detection and tracking precision, which might be explained by the inconsistency of the cell mark annotations mentioned in Section 2.4.

Table 5.4 shows a comparison of our tracking pipeline using our data association algorithm and a modified version using the approach by Bise, Yin and Kanade (2011) for computing the final tracks. We note that the proposed modifications only slightly improve the DET and TRA metrics for most datasets, but they provide a significant reduction in the number of generated hypotheses. As a consequence, it allows faster solution computation and fewer hardware requirements.

Finally, we provide visual detection results on the CTC (MAŠKA et al., 2014) datasets on Figure 5.1. We can observe that EBBs provide a good description of the cell shapes for the GOWT1 and HeLa datasets, but not so much for the U373 datasets. We can also note the high recall rate of the object detector, since almost all cells are retrieved. Figure 5.2 presents the generated tracking trees[§]. Analyzing the results for the GOWT1 and U373 datasets, which are less cluttered, we can observe that our method could produce clear paths for most of the initial cells. These datasets have almost no mitosis or apoptosis events, so the paths that seem to emerge in later frames can be false positives or cells emerging from the image borders.

5.3 Sensitivity analysis

In this section, we analyze the sensitivity of our tracking-by-detection method regarding its hyper-parameters. We randomly sampled the parameter values within an interval and evaluated the results on the CTC (MAŠKA et al., 2014) tested training datasets (i.e., with provided GT). The parameters were randomly sampled in the following scheme: λ_{link} and $\lambda_{mit} \in \mathbb{R}^+$ linearly spaced between 5 and 1000 with step 25, $t_{th} \in \mathbb{N}$ linearly ranging from 1 to 8 with step 1, $\tau_s \in \mathbb{R}^+$ linearly ranging from 0.4 to 0.9 with 0.05 step, and $\tau_{FP} \in \mathbb{R}^+$ linearly ranging from 0.5 to 0.9 with 0.05 step. For consistently evaluating the impact of the parameters on the different datasets, we subtracted the metrics values

[§]We also provide animated images in our GitHub repository at: <https://github.com/LucasKirsten/Deep-Cell-Tracking-EBB/>

from the ones reported in Table 5.1 when using the watershed method to show the relative gain or loss when changing the hyper-parameters.

Figure 5.3 shows a boxplot with the relative changes of the DET and TRA metrics for a set of ~ 400 random combinations of the hyper-parameters in each individual dataset. We can observe that the impact of changing the parameters is small for most of the datasets. The worst-case scenario occurs on the U373-2 dataset, with a negative impact of $\sim 3.5\%$ on the TRA metric. On the other hand, we note that some combination of hyper-parameters can actually improve the results obtained with the default parameters.

In order to access the impact of the individual hyper-parameters on the method, we used the Shapley Additive Explanations (SHAP values) (LUNDBERG; LEE, 2017; LUNDBERG et al., 2018) using the DET and TRA evaluation metrics as the targets and averaging the results among all the CTC tested datasets. SHAP values provide insights into feature contributions, distributing credit among features to explain machine learning predictions. They measure the impact of each feature compared to its absence or average value, allowing a nuanced understanding of feature importance. Positive values increase predictions, negative values decrease predictions, and zero values have no impact. Examining SHAP values helps identify influential features, aiding feature selection, model debugging, and understanding model decisions. In our case, we adapted the method to work with the hyper-parameters instead of the features. Figure 5.4 presents the violin plots, where the color indicates the parameter value and the horizontal axis denotes the corresponding SHAP value. The small range of the horizontal axis and concentration at small SHAP values indicate that the method is robust to the parameter choice. Furthermore, we note that most random combinations of individual parameters lead to a positive impact on the metric scores.

Table 5.1 – Results for the CTC (MAŠKA et al., 2014) training datasets using separate sequences. Our results with the watershed method are reported as *Ours-W*. **Bold** values mark the best results, while underline values mark the second best. We also report the annotation requirements (Ann. Req. column) of each technique related to the detection (Det) and tracking (Tra)

Dataset	Method	Ann. Req.		DET	SEG	EBB SEG	TRA
		Det.	Tra.				
GOWT1-01	ST-TCV	✗	✗	*N/A*	*N/A*	-	0.913
	KTH	✗	✗	*N/A*	0.6849	-	0.9462
	BLOB	✗	✓	*N/A*	0.7415	-	0.9733
	CPN	✓	✓	*N/A*	0.8506	-	0.9864
	DRL	✓	✓	*N/A*	**0.8585**	-	0.9875
	Ours	✓	✗	0.9916	0.8568	0.9268	0.9914
	Ours-W	✓	✗	0.9940	-	-	**0.9930**
GOWT1-02	ST-TCV	✗	✗	*N/A*	*N/A*	-	0.914
	HEID	✓	✓	*N/A*	*N/A*	-	0.95
	EPFL	✓	✓	*N/A*	*N/A*	-	0.95
	KTH	✗	✗	*N/A*	0.8942	-	0.9452
	BLOB	✗	✓	*N/A*	0.9046	-	0.9628
	CPN	✓	✓	*N/A*	0.8725	-	0.9719
	DRL	✓	✓	*N/A*	**0.9181**	-	0.9575
	Ours	✓	✗	0.9812	0.8509	0.9167	0.9817
	Ours-W	✓	✗	0.9868	-	-	**0.9853**
U373-01	CPN	✓	✓	*N/A*	0.7336	-	0.9594
	DRL	✓	✓	*N/A*	**0.8527**	-	**0.9919**
	Ours	✓	✗	0.9647	0.6307	0.7791	0.9671
	Ours-W	✓	✗	0.9748	-	-	0.9774
U373-02	CPN	✓	✓	*N/A*	0.7376	-	0.9346*
	DRL	✓	✓	*N/A*	**0.7735**	-	0.9318
	Ours	✓	✗	0.8822	0.5626	0.7029	0.8737
	Ours-W	✓	✗	0.9634	-	-	**0.9525**
HeLa-01	ST-TCV	✗	✗	*N/A*	*N/A*	-	0.816
	HEID	✓	✓	*N/A*	*N/A*	-	0.80
	EPFL	✓	✓	*N/A*	*N/A*	-	0.98
	KTH	✗	✗	*N/A*	0.8018	-	0.9775
	CPN	✓	✓	*N/A*	**0.8313**	-	**0.9869**
	BLOB	✗	✓	*N/A*	0.7951	-	0.9803
	Ours	✓	✗	0.9779	0.7264	0.8871	0.9758
	Ours-W	✓	✗	0.9863	-	-	0.9820
HeLa-02	ST-TCV	✗	✗	*N/A*	*N/A*	-	0.845
	HEID	✓	✓	*N/A*	*N/A*	-	0.85
	EPFL	✓	✓	*N/A*	*N/A*	-	0.97
	KTH	✗	✗	*N/A*	0.8366	-	0.9747
	CPN	✓	✓	*N/A*	**0.8445**	-	**0.9826**
	BLOB	✗	✓	*N/A*	0.8390	-	0.9771
	Ours	✓	✗	0.9707	0.7618	0.8897	0.9664
	Ours-W	✓	✗	0.9796	-	-	0.9740

* denotes augmentation on the segmentation masks.

Source: The authors.

Table 5.2 – Results from the CTC (MAŠKA et al., 2014) challenge evaluation server. We report the results for our technique both using and not the watershed mask augmentation (*W* column), the rank position over all submissions, and the relative difference to the first rank method (*To TOP1* column). Evaluation date: October 10, 2022.

Dataset	W	DET			TRA		
		Score	Rank	To TOP1	Score	Rank	To TOP1
GOWT1	✗	0.925	26/49	5.61%	0.922	19/40	5.82%
	✓	0.970	3/49	1.02%	0.959	4/40	2.04%
U373	✗	0.914	33/38	7.68%	0.909	26/31	7.72%
	✓	0.979	17/38	1.11%	0.976	12/31	0.91%
HeLa	✗	0.986	15/48	0.80%	0.984	11/39	0.91%
	✓	0.989	10/48	0.50%	0.988	8/39	0.50%

Source: The authors.

Table 5.3 – Results using standard detection and tracking quality metrics (in %), as described in Section 2.4.

Dataset	Tracking						Detection			
	ID-F1	ID-P	ID-R	R	P	MOTA	R_{50}	P_{50}	$F1_{50}$	AP_{50}
GOWT1-01	97.9	96.7	99.0	99.9	97.5	97.0	99.4	97.0	98.2	90.9
GOWT1-02	95.2	91.3	99.6	100.0	91.7	90.7	99.9	83.7	91.1	90.5
U373-01	88.9	79.9	100.0	100.0	79.9	74.8	99.9	79.0	88.2	90.8
U373-02	81.5	79.6	83.6	96.0	91.4	86.3	93.6	90.1	91.8	89.5
HeLa-01	94.6	92.4	96.8	100.0	95.4	94.0	90.8	96.5	93.6	90.4
HeLa-02	93.1	89.0	97.6	100.0	91.2	89.1	95.0	93.5	94.2	90.1

Source: The authors.

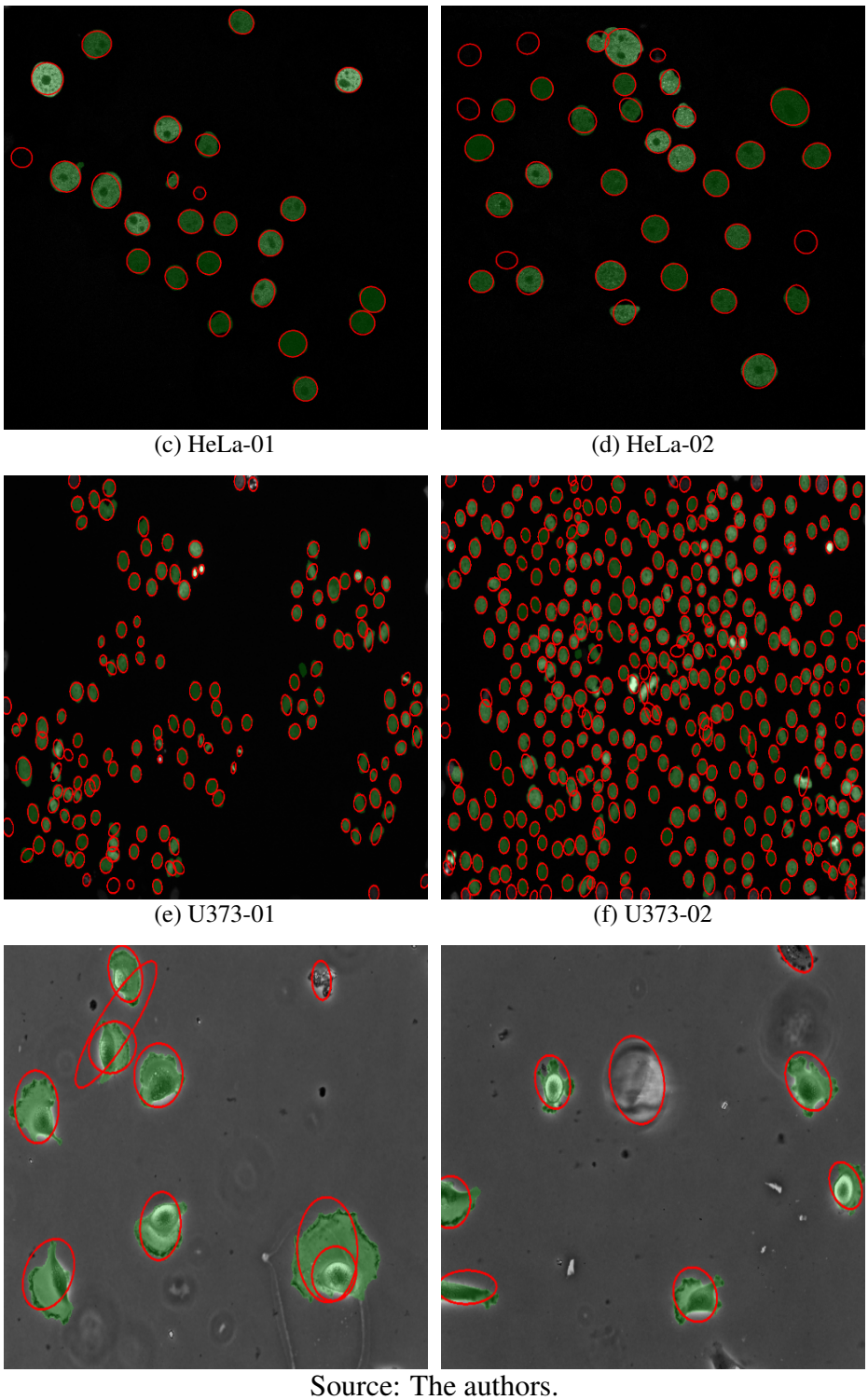
Table 5.4 – Results comparing our complete method and a modified version of Bise, Yin and Kanade (2011) (Baseline).

Dataset	Method	DET	TRA	Hypothesis	Time (s)
GOWT1-01	Baseline	0.9914	0.9913	133	0.1234
	Ours	0.9916	0.9914	47	0.0998
GOWT1-02	Baseline	0.9803	0.9805	312	0.2280
	Ours	0.9812	0.9817	161	0.2082
U373-01	Baseline	0.9655	0.9677	78	0.0891
	Ours	0.9647	0.9671	53	0.0763
U373-02	Baseline	0.8818	0.8733	83	0.0918
	Ours	0.8822	0.8737	24	0.0509
HeLa-01	Baseline	0.9772	0.9741	7989	30.681
	Ours	0.9779	0.9758	7061	25.337
HeLa-02	Baseline	0.9699	0.9652	25525	259.18
	Ours	0.9707	0.9664	22649	235.06

Source: The authors.

Figure 5.1 – Visual results of our method in the CTC (MAŠKA et al., 2014) training datasets. In green are the ground-truth segmentation masks, and in red are the predicted EBBs.

(a) GOWT1-01 (b) GOWT1-02



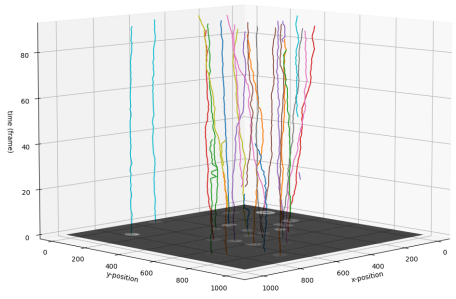
(c) HeLa-01 (d) HeLa-02

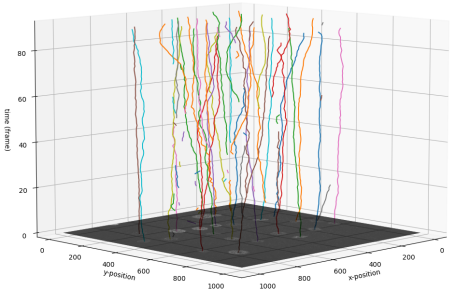
(e) U373-01 (f) U373-02

Source: The authors.

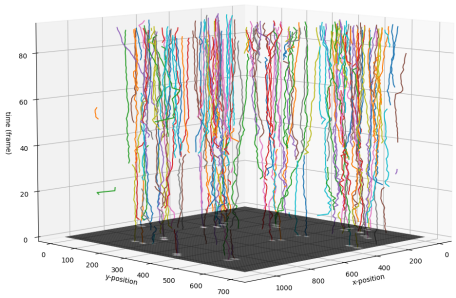
Figure 5.2 – Visualization of the generated tracking trees in the CTC (MAŠKA et al., 2014) training datasets.

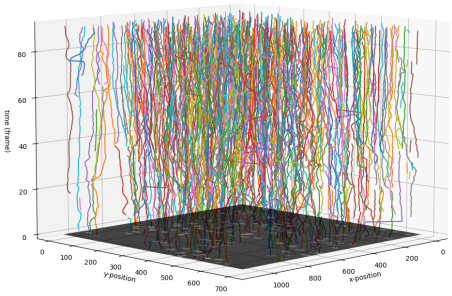
(a) GOWT1-01 (b) GOWT1-02



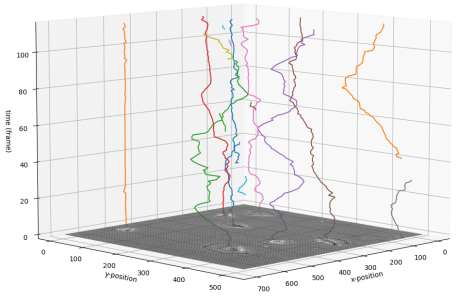


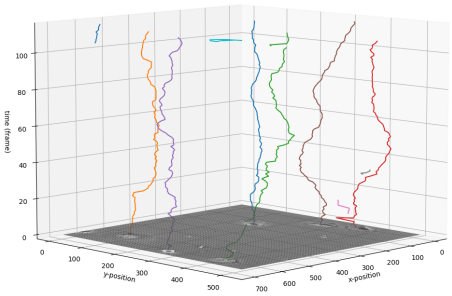
(c) HeLa-01 (d) HeLa-02





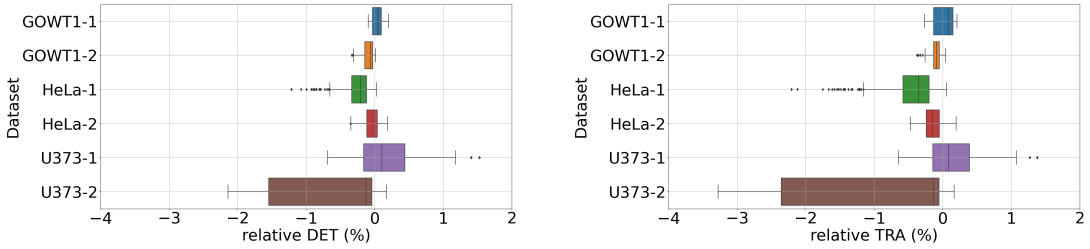
(e) U373-01 (f) U373-02





Source: The authors.

Figure 5.3 – Impacts on the individual datasets evaluation from randomly sampling the hyper-parameters. The zero value refers to an output metric value equal to the one reported in Table 5.1.

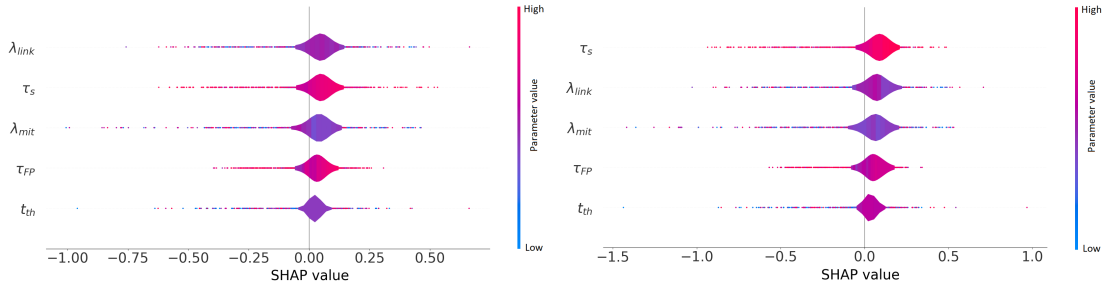


Source: The authors.

Figure 5.4 – Individual impacts of the hyper-parameters on the method performance using the SHAP values (LUNDBERG; LEE, 2017; LUNDBERG et al., 2018) regarding the DET and TRA metrics.

(a) DET metric

(b) TRA metric



Source: The authors.

6 CONCLUSIONS

In this thesis, we proposed a tracking-by-detection method that explores an OBB detector to identify cells, represents them as ellipses, and then uses the detection information in an unsupervised tracking algorithm based on tracklet association. Our method alleviates the annotation efforts by representing the cells as a 5-parameter oriented ellipse (that can be either annotated as an OBB or EBB), and by defining an unsupervised tracking system oriented solely on the detection information retrieved by the trained object detector.

Our results demonstrate that general-purpose oriented object detectors are usually suitable for detecting cells in such representation. Moreover, by leveraging the easy mapping between OBBs to EBBs, the cell elliptical representation presents a good approximation for the full segmentation masks, particularly for lineages with a regular shape. The EBB representation further relates to the GBB representation, which enables the usage of the Hellinger distance for computing region similarities and is closely related to the IoU metric. This allowed us to directly use such distance for both the NMS and the tracklet association algorithms.

Furthermore, our tracking-by-detection method can achieve results competitive to other SOTA methods that require considerably more annotated data. It reduces the hardware requirements for training and predicting when compared to the current trend of end-to-end trackers, since it requires training only one object detector and does not rely on training a complete detection and association deep learning architecture that needs both batches of frame images and their objects associations. Moreover, although it relies on "tunable" parameters, we demonstrate that, in general, their choice causes a small impact on the detection and tracking results. We believe that our method can be broadly used in applications where there are limited resources or short deadlines for retrieving the full annotations.

In future works, we intend to further investigate methods to improve both the object detector and the complete tracking system. For instance, instead of using the standard smooth-ℓ_1 loss for regressing the OBB parameters in the object detectors training, we could regress it using the Helinger distance or other probabilistic loss function directly, as done by Llerena et al. (2021), Yang et al. (2021a), Yang et al. (2021b). Furthermore, it is possible to adjust certain hyper-parameters of the network to be more suitable for each dataset (e.g., the anchors). For the tracking system, we could investigate strategies that use the cell movements to generate "reliable" tracklets and then compute the likelihoods for

the global data association step, such as the modified Kalman filter proposed by Li et al. (2008). Finally, recent works have employed techniques related to few-shot (WANG et al., 2020b), semi-supervised (YANG et al., 2022) and self-learning (LIU et al., 2021) in order to alleviate even more the data annotation requirements.

During the course of this work, we have submitted the following papers:

- Cell Tracking-by-detection using Elliptical Bounding Boxes (2022) – IEEE/ACM Transactions on Computational Biology and Bioinformatics (Submitted).
- Probabilistic Intersection-over-Union for Training and Evaluation of Oriented Object Detectors (2022) – IEEE Transactions on Image Processing (Submitted). Pre-print publication: (LLERENA et al., 2021).
- Can We Trust Bounding Box Annotations for Object Detection? (2022) – IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) Workshops (MURRUGARRA-LLERENA; KIRSTEN; JUNG, 2022)

REFERENCES

AKRAM, S. U. et al. Joint cell segmentation and tracking using cell proposals. In: IEEE. **2016 IEEE 13th International Symposium on Biomedical Imaging (ISBI)**. [S.l.], 2016. p. 920–924.

AKRAM, S. U. et al. Cell tracking via proposal generation and selection. **arXiv preprint arXiv:1705.03386**, 2017.

ANOSHINA, N. A.; SOROKIN, D. V. Weak supervision using cell tracking annotation and image registration improves cell segmentation. In: IEEE. **2022 Eleventh International Conference on Image Processing Theory, Tools and Applications (IPTA)**. [S.l.], 2022. p. 1–5.

BENSCH, R.; RONNEBERGER, O. Cell segmentation and tracking in phase contrast images using graph cut with asymmetric boundary costs. In: IEEE. **2015 IEEE 12th International Symposium on Biomedical Imaging (ISBI)**. [S.l.], 2015. p. 1220–1223.

BERNARDIN, K.; STIEFELHAGEN, R. Evaluating multiple object tracking performance: the clear mot metrics. **EURASIP Journal on Image and Video Processing**, Springer, v. 2008, p. 1–10, 2008.

BHATTACHARYYA, A. On a measure of divergence between two multinomial populations. **Sankhyā: the indian journal of statistics**, JSTOR, p. 401–406, 1946.

BISE, R.; YIN, Z.; KANADE, T. Reliable cell tracking by global data association. In: IEEE. **2011 IEEE International Symposium on Biomedical Imaging: From Nano to Macro**. [S.l.], 2011. p. 1004–1010.

BOCHKOVSKIY, A.; WANG, C.-Y.; LIAO, H.-Y. M. Yolov4: Optimal speed and accuracy of object detection. **arXiv preprint arXiv:2004.10934**, 2020.

BOUKARI, F.; MAKROGIANNIS, S. Joint level-set and spatio-temporal motion detection for cell segmentation. **BMC Medical Genomics**, BioMed Central, v. 9, n. 2, p. 179–194, 2016.

BOUKARI, F.; MAKROGIANNIS, S. Automated cell tracking using motion prediction-based matching and event handling. **IEEE/ACM transactions on computational biology and bioinformatics**, IEEE, v. 17, n. 3, p. 959–971, 2018.

BRADSKI, G. The OpenCV Library. **Dr. Dobb's Journal of Software Tools**, 2000.

CARION, N. et al. End-to-end object detection with transformers. In: SPRINGER. **European conference on computer vision**. [S.l.], 2020. p. 213–229.

CHEN, Z. et al. Piou loss: Towards accurate oriented object detection in complex environments. In: SPRINGER. **European conference on computer vision**. [S.l.], 2020. p. 195–211.

DENG, J. et al. Imagenet: A large-scale hierarchical image database. In: IEEE. **2009 IEEE conference on computer vision and pattern recognition**. [S.l.], 2009. p. 248–255.

Ding, J. et al. Learning roi transformer for oriented object detection in aerial images. In: **CVPR**. [S.l.: s.n.], 2019. p. 2844–2853.

EMAMI, N.; SEDAEI, Z.; FERDOUSI, R. Computerized cell tracking: Current methods, tools and challenges. **Visual Informatics**, Elsevier, v. 5, n. 1, p. 1–13, 2021.

FORREST, J. et al. **coin-or/Cbc: Release releases/2.10.8**. Zenodo, 2022. Available from Internet: <https://doi.org/10.5281/zenodo.6522795>.

GIUSEPPE, D. D. et al. Learning cancer-related drug efficacy exploiting consensus in coordinated motility within cell clusters. **IEEE Transactions on Biomedical Engineering**, IEEE, v. 66, n. 10, p. 2882–2888, 2019.

GRADECI, D. et al. Single-cell approaches to cell competition: high-throughput imaging, machine learning and simulations. In: ELSEVIER. **Seminars in cancer biology**. [S.l.], 2020. v. 63, p. 60–68.

GUPTA, D. K. et al. Tracking-assisted segmentation of biological cells. **arXiv preprint arXiv:1910.08735**, 2019.

HAYASHIDA, J.; NISHIMURA, K.; BISE, R. Consistent cell tracking in multi-frames with spatio-temporal context by object-level warping loss. In: **Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision**. [S.l.: s.n.], 2022. p. 1727–1736.

HE, K. et al. Mask r-cnn. In: **Proceedings of the IEEE international conference on computer vision**. [S.l.: s.n.], 2017. p. 2961–2969.

HE, K. et al. Deep residual learning for image recognition. In: **Proceedings of the IEEE conference on computer vision and pattern recognition**. [S.l.: s.n.], 2016. p. 770–778.

He, S.; Lau, R. W. H. Oriented object proposals. In: **ICCV**. [S.l.: s.n.], 2015. p. 280–288.

HELLINGER, E. Neue begründung der theorie quadratischer formen von unendlichvielen veränderlichen. **Journal für die reine und angewandte Mathematik**, De Gruyter, v. 1909, n. 136, p. 210–271, 1909.

HIROSE, O. et al. Spf-celltracker: Tracking multiple cells with strongly-correlated moves using a spatial particle filter. **IEEE/ACM transactions on computational biology and bioinformatics**, IEEE, v. 15, n. 6, p. 1822–1831, 2017.

HUH, S. et al. Automated mitosis detection of stem cell populations in phase-contrast microscopy images. **IEEE transactions on medical imaging**, IEEE, v. 30, n. 3, p. 586–596, 2010.

JIANG, Y. et al. R2cnn: Rotational region cnn for orientation robust scene text detection. **arXiv preprint arXiv:1706.09579**, 2017.

KAILATH, T. The divergence and bhattacharyya distance measures in signal selection. **IEEE transactions on communication technology**, IEEE, v. 15, n. 1, p. 52–60, 1967.

KARATZAS, D. et al. Icdar 2015 competition on robust reading. In: **2015 13th International Conference on Document Analysis and Recognition (ICDAR)**. [S.l.: s.n.], 2015. p. 1156–1160.

KÖRBER, N. Mia: An open source standalone deep learning application for microscopic image analysis. **bioRxiv**, Cold Spring Harbor Laboratory, 2022.

KUHN, H. W. The hungarian method for the assignment problem. **Naval research logistics quarterly**, Wiley Online Library, v. 2, n. 1-2, p. 83–97, 1955.

KULHARIA, V. et al. Box2seg: Attention weighted loss and discriminative feature learning for weakly supervised segmentation. In: SPRINGER. **European Conference on Computer Vision**. [S.l.], 2020. p. 290–308.

LEITE, M. R. C.; CESTARI, I. A.; CESTARI, I. N. Computational tool for morphological analysis of cultured neonatal rat cardiomyocytes. In: IEEE. **2015 37th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC)**. [S.l.], 2015. p. 3517–3520.

LI, K. et al. Cell population tracking and lineage construction with spatiotemporal context. **Medical image analysis**, Elsevier, v. 12, n. 5, p. 546–566, 2008.

LIN, T.-Y. et al. Feature pyramid networks for object detection. In: **Proceedings of the IEEE conference on computer vision and pattern recognition**. [S.l.: s.n.], 2017. p. 2117–2125.

LIN, T.-Y. et al. Focal loss for dense object detection. In: **Proceedings of the IEEE international conference on computer vision**. [S.l.: s.n.], 2017. p. 2980–2988.

LIU, D. et al. Panoptic feature fusion net: a novel instance segmentation paradigm for biomedical and biological images. **IEEE Transactions on Image Processing**, IEEE, v. 30, p. 2045–2059, 2021.

LIU, L. et al. Deep learning for generic object detection: A survey. **International journal of computer vision**, Springer, v. 128, n. 2, p. 261–318, 2020.

LIU, W. et al. Ssd: Single shot multibox detector. In: SPRINGER. **European conference on computer vision**. [S.l.], 2016. p. 21–37.

LIU, X. et al. Self-supervised learning: Generative or contrastive. **IEEE Transactions on Knowledge and Data Engineering**, IEEE, v. 35, n. 1, p. 857–876, 2021.

LIU, Z. et al. A high resolution optical satellite image dataset for ship recognition and some new baselines. In: INSTICC. **the 6th International Conference on Pattern Recognition Applications and Methods - Volume 1: ICPRAM,**. [S.l.]: SciTePress, 2017. p. 324–331. ISBN 978-989-758-222-6.

LLERENA, J. M. et al. Gaussian bounding boxes and probabilistic intersection-over-union for object detection. **arXiv preprint arXiv:2106.06072**, 2021.

LU, K. et al. Biofabrication of aligned structures that guide cell orientation and applications in tissue engineering. **Bio-Design and Manufacturing**, Springer, v. 4, n. 2, p. 258–277, 2021.

LUNDBERG, S. M.; LEE, S.-I. A unified approach to interpreting model predictions. In: GUYON, I. et al. (Ed.). **Advances in Neural Information Processing Systems 30**. Curran Associates, Inc., 2017. p. 4765–4774. Available from Internet: <http://papers.nips.cc/paper/7062-a-unified-approach-to-interpreting-model-predictions.pdf>.

LUNDBERG, S. M. et al. Explainable machine-learning predictions for the prevention of hypoxaemia during surgery. **Nature Biomedical Engineering**, Nature Publishing Group, v. 2, n. 10, p. 749, 2018.

MAGNUSSON, K. E.; JALDÉN, J. A batch algorithm using iterative application of the viterbi algorithm to track cells and construct cell lineages. In: IEEE. **2012 9th IEEE International Symposium on Biomedical Imaging (ISBI)**. [S.l.], 2012. p. 382–385.

MANDAL, S.; UHLMANN, V. Splinedist: Automated cell segmentation with spline curves. In: IEEE. **2021 IEEE 18th International Symposium on Biomedical Imaging (ISBI)**. [S.l.], 2021. p. 1082–1086.

MAŠKA, M. et al. A benchmark for comparison of cell tracking algorithms. **Bioinformatics**, Oxford University Press, v. 30, n. 11, p. 1609–1617, 2014.

MILAN, A. et al. Mot16: A benchmark for multi-object tracking. **arXiv preprint arXiv:1603.00831**, 2016.

MURRUGARRA-LLERENA, J.; KIRSTEN, L. N.; JUNG, C. R. Can we trust bounding box annotations for object detection? In: **Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) Workshops**. [S.l.: s.n.], 2022. p. 4813–4822.

NAYEF, N. et al. Icdar2017 robust reading challenge on multi-lingual scene text detection and script identification - rrc-mlt. In: **2017 14th IAPR International Conference on Document Analysis and Recognition (ICDAR)**. [S.l.: s.n.], 2017. v. 01, p. 1454–1459.

NEWELL, A.; HUANG, Z.; DENG, J. Associative embedding: End-to-end learning for joint detection and grouping. In: **Advances in neural information processing systems**. [S.l.: s.n.], 2017. p. 2277–2287.

NISHIMURA, K. et al. Weakly-supervised cell tracking via backward-and-forward propagation. In: SPRINGER. **European Conference on Computer Vision**. [S.l.], 2020. p. 104–121.

OH, H.-J.; LEE, K.; JEONG, W.-K. Scribble-supervised cell segmentation using multiscale contrastive regularization. In: IEEE. **2022 IEEE 19th International Symposium on Biomedical Imaging (ISBI)**. [S.l.], 2022. p. 1–5.

PARK, Y. et al. Multiple object tracking in deep learning approaches: A survey. **Electronics**, MDPI, v. 10, n. 19, p. 2406, 2021.

PAYER, C. et al. Segmenting and tracking cell instances with cosine embeddings and recurrent hourglass networks. **Medical image analysis**, Elsevier, v. 57, p. 106–119, 2019.

PAYER, C. et al. Instance segmentation and tracking with cosine embeddings and recurrent hourglass networks. In: SPRINGER. **International Conference on Medical Image Computing and Computer-Assisted Intervention**. [S.l.], 2018. p. 3–11.

QIAN, W. et al. Learning modulated loss for rotated object detection. **arXiv preprint arXiv:1911.08299**, 2019.

REDMON, J. et al. You only look once: Unified, real-time object detection. In: **Proceedings of the IEEE conference on computer vision and pattern recognition**. [S.l.: s.n.], 2016. p. 779–788.

REN, S. et al. Faster r-cnn: towards real-time object detection with region proposal networks. **IEEE transactions on pattern analysis and machine intelligence**, IEEE, v. 39, n. 6, p. 1137–1149, 2016.

RISTANI, E. et al. Performance measures and a data set for multi-target, multi-camera tracking. In: SPRINGER. **European conference on computer vision**. [S.l.], 2016. p. 17–35.

RONNEBERGER, O.; FISCHER, P.; BROX, T. U-net: Convolutional networks for biomedical image segmentation. In: SPRINGER. **International Conference on Medical image computing and computer-assisted intervention**. [S.l.], 2015. p. 234–241.

ROSSUM, G. V.; JR, F. L. D. **Python reference manual**. [S.l.]: Centrum voor Wiskunde en Informatica Amsterdam, 1995.

SCHMIDT, U. et al. Cell detection with star-convex polygons. In: SPRINGER. **International Conference on Medical Image Computing and Computer-Assisted Intervention**. [S.l.], 2018. p. 265–273.

SIMONYAN, K.; ZISSERMAN, A. Very deep convolutional networks for large-scale image recognition. **arXiv preprint arXiv:1409.1556**, 2014.

SYED, T. Q. et al. Detection and counting of" in vivo" cells to predict cell migratory potential. In: IEEE. **2008 First Workshops on Image Processing Theory, Tools and Applications**. [S.l.], 2008. p. 1–8.

TAN, M.; PANG, R.; LE, Q. V. Efficientdet: Scalable and efficient object detection. In: **Proceedings of the IEEE/CVF conference on computer vision and pattern recognition**. [S.l.: s.n.], 2020. p. 10781–10790.

TÜRETKEN, E. et al. Globally optimal cell tracking using integer programming. **arXiv preprint arXiv:1501.05499**, 2015.

ULMAN, V. et al. An objective comparison of cell-tracking algorithms. **Nature methods**, Nature Publishing Group, v. 14, n. 12, p. 1141–1152, 2017.

WANG, C.-Y.; BOCHKOVSKIY, A.; LIAO, H.-Y. M. Scaled-yolov4: Scaling cross stage partial network. In: **Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)**. [S.l.: s.n.], 2021. p. 13029–13038.

WANG, J. et al. Deep reinforcement learning for data association in cell tracking. **Frontiers in Bioengineering and Biotechnology**, Frontiers Media SA, v. 8, p. 298, 2020.

WANG, Y. et al. Generalizing from a few examples: A survey on few-shot learning. **ACM computing surveys (csur)**, ACM New York, NY, USA, v. 53, n. 3, p. 1–34, 2020.

XIA, G.-S. et al. Dota: A large-scale dataset for object detection in aerial images. In: **The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)**. [S.l.: s.n.], 2018.

XU, B. et al. An automated cell tracking approach with multi-bernoulli filtering and ant colony labor division. **IEEE/ACM transactions on computational biology and bioinformatics**, IEEE, v. 18, n. 5, p. 1850–1863, 2019.

YANG, J.; LI, C.; GAO, J. Focal modulation networks. **arXiv preprint arXiv:2203.11926**, 2022.

YANG, X. et al. Dense label encoding for boundary discontinuity free rotation detection. In: **Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition**. [S.l.: s.n.], 2021. p. 15819–15829.

YANG, X. et al. A survey on deep semi-supervised learning. **IEEE Transactions on Knowledge and Data Engineering**, IEEE, 2022.

YANG, X. et al. R3det: Refined single-stage detector with feature refinement for rotating object. In: **AAAI**. [S.l.: s.n.], 2021.

YANG, X.; YAN, J.; HE, T. On the arbitrary-oriented object detection: Classification based approaches revisited. **arXiv preprint arXiv:2003.05597**, 2020.

YANG, X. et al. Rethinking rotated object detection with gaussian wasserstein distance loss. In: **International Conference on Machine Learning (ICML)**. [S.l.: s.n.], 2021.

YANG, X. et al. Learning high-precision bounding box for rotated object detection via kullback-leibler divergence. In: RANZATO, M. et al. (Ed.). **Advances in Neural Information Processing Systems**. Curran Associates, Inc., 2021. v. 34, p. 18381–18394. Available from Internet: <https://proceedings.neurips.cc/paper/2021/file/98f13708210194c475687be6106a3b84-Paper.pdf>.

ZAIDI, S. S. A. et al. A survey of modern deep learning based object detection models. **Digital Signal Processing**, Elsevier, p. 103514, 2022.

ZHAO, M. et al. Faster mean-shift: Gpu-accelerated clustering for cosine embedding-based cell segmentation and tracking. **Medical Image Analysis**, Elsevier, v. 71, p. 102048, 2021.

ZHAO, T.; YIN, Z. Weakly supervised cell segmentation by point annotation. **IEEE Transactions on Medical Imaging**, IEEE, v. 40, n. 10, p. 2736–2747, 2020.

ZHAO, Z. et al. Deep learning based instance segmentation in 3d biomedical images using weak annotation. In: SPRINGER. **International Conference on Medical Image Computing and Computer-Assisted Intervention**. [S.l.], 2018. p. 352–360.

APPENDIX A — RESUMO EXPANDIDO

A detecção e o rastreamento de células vivas em imagens de microscopia são tarefas críticas em aplicações biomédicas. No entanto, a enorme quantidade de dados produzidas em tais aplicações gera um desafio para o seu uso, que só pode ser superado com ferramentas computacionais apropriadas. As abordagens de aprendizado profundo são o estado-da-arte para detecção e rastreamento de células, onde a representação do formato das células tem um impacto significativo tanto no tempo dedicado à anotação de imagens quanto na complexidade da rede que deve ser empregada. Por exemplo, o uso de máscaras de segmentação pode impor problemas relacionados à anotação dos dados, já que cada pixel da imagem precisa ser anotado; por outro lado, o uso das tradicionais caixas horizontais (*Horizontal Bounding Boxes*, ou HBBs) pode não representar bem o formato das células, já que elas podem conter grande parte do fundo da imagem.

Neste trabalho, nós propomos um algoritmo de rastreamento-por-detecção de células que representa as células como caixas elípticas orientadas (*Elliptical Bounding Boxes*, EBB), que demonstra um bom compromisso entre simplicidade e completude. O método proposto requer apenas anotações de células como caixas orientadas (i.e., requer apenas o ponto central x e y, a largura, altura e orientação da célula na imagem), e nenhuma anotação de rastreamento envolvendo sequências temporais é necessária (e.g., associação entre células de quadros subsequentes). Mais especificamente, o método proposto usa um modelo de aprendizado profundo para detectar as células como caixas orientadas (*Oriented Bounding Boxes*, OBBs) e, em seguida, converte-as em EBBs para ajustar melhor a forma das células. O algoritmo de rastreamento é baseado em um algoritmo de associação global de dados de longo prazo não-supervisionado, que depende apenas das informações de detecção fornecidas pelo detector de objetos orientados (e.g., formato e posição das células, lapso temporal entre detecções).

Nosso método consiste em três etapas principais: (i) detecção das células, (ii) geração de trechos curtos de trajetória e (iii) associação global de dados para obter as trajetórias finais das células e as árvores de linhagem. Na etapa (i), um detector de objetos orientados de propósito geral é utilizados para identificar células em cada quadro. A saída do detector de objetos é na forma de caixas delimitadoras orientadas (OBBs), que são então convertidas em caixas delimitadoras elípticas (EBBs) para melhor adaptar o formato das células. Nós também propomos um algoritmo de supressão de não-máximos (*Non-maximum suppression*, NMS) que mapeia a representação das células de EBB para uma

distribuição gaussiana bidimensional a fim de usar da distância de Helinger para eliminar possíveis detecções com super-posição (i.e., detecções repetidas para uma mesma célula), assim mitigando o cálculo complexo de intersecção sobre a união (*Intersection over Union*, IoU) para EBBs e eliminando o problema de ambiguidade em relação a orientação do objeto (e.g., objetos circulares sem orientação definida). Essa mesma técnica de mapeamento é utilizada para calcular a sobreposição de células em quadros subsequentes a fim de gerar os trechos curtos (a.k.a. *tracklets*) para a etapa (ii), onde assumimos pouco movimento e mudança no formato das células. As trilhas curtas são então associadas usando um método de associação de dados global (iii) baseado no trabalho de Bise, Yin and Kanade (2011). Para tal, nós propomos as seguintes modificações ao algoritmo original:

- Não diferenciar entre células mitóticas e não mitóticas;
- Remover as hipóteses de iniciação e término, e as substituir por uma hipótese de completude para simplificar o problema de *maximum-a-posteriori* (MAP) que o algoritmo soluciona;
- Redefinir todos os cálculos de probabilidade para usar apenas informações sobre as células detectadas (e.g., valor da confiança de detecção para discriminar entre hipóteses verdadeiras e falsas positivas, posição da célula na imagem).

Em nossos experimentos, nós empregamos dois conjuntos diferentes de dados: um conjunto privado de dados de células anotados como OBBs (chamado de dados de células orientadas – *Oriented Cell Dataset*, OCD), e três conjuntos de dados públicos do desafio de rastreamento celular (*Cell Tracking Challenge*, CTC (MAŠKA et al., 2014)). O OCD é composto principalmente por células de glioblastoma. Tais imagens apresentam baixo contraste em relação ao fundo, o que torna muito difícil segmentar as células individuais de forma consistente. Além disso, a forma e o tamanho das células de glioblastoma podem variar consideravelmente, mas elas têm uma forma geralmente alongada. Usar HBBs para detectá-las não é uma boa escolha, pois a HBB de uma célula pode não capturar a forma e alongamento real, enquanto também pode conter células próximas e possivelmente grandes porções do fundo. Esse conjunto foi utilizado para determinar o detector orientado mais adequado para se utilizar num cenário de propósito geral (e.g., como o conjunto do CTC). Nós treinamos e avaliamos sete detectores de propósito geral e verificamos que o mais adequado seria o R^2CNN (JIANG et al., 2017), pois ele mostrou, em geral, um desempenho superior quando comparado aos outros nas métricas de precisão média (*Average Precision*, ou AP), precisão, revocação (*recall*) e F1-score.

Nós utilizamos os seguintes conjuntos de imagens do desafio CTC: Fluo-N2DH-GOWT1 (composto por células troncos embrionárias de ratos em fluorescência), PhC-C2DH-U373 (composto por células de glioblastoma-astrocitoma em contraste de fase) e Fluo-N2DH-HeLa (composto por imagens fluorescentes de núcleos). Para cada conjunto de dados, há dois conjuntos de linhagens: um de treinamento (com anotações), e um de teste (sem anotações). Nós utilizamos os dados de treinamento para determinar os parâmetros do nosso rastreador e submetemos nosso algoritmo para o servidor do CTC para que ele fosse avaliado e comparado a outros métodos de acordo com as métricas DET, SEG e TRA propostas por Maška et al. (2014) nos dados de teste.

Nossos resultados demonstram que a representação elíptica da célula apresenta uma boa aproximação para a máscara de segmentação completa, especialmente para linhagens com forma regular. Além disso, nosso método de rastreamento por detecção pode alcançar resultados competitivos em relação a outros métodos de última geração que requerem consideravelmente mais dados anotados. Outrossim, nosso método reduz os requisitos de hardware para treinamento e predição em comparação com a tendência atual de rastreadores de ponta a ponta, uma vez que requer apenas um detector de objetos para treinamento e não depende do treinamento de uma arquitetura completa de detecção e associação de aprendizado profundo que necessita de ambos quadros de imagens e as associações de objetos entre os mesmos. Acreditamos que nosso método possa ser amplamente utilizado em aplicações onde existam recursos limitados ou prazos curtos para se obter as anotações completas.