Medical Image Analysis 57 (2019) 106–119

Contents lists available at ScienceDirect

Medical Image Analysis

journal homepage: www.elsevier.com/locate/media

Segmenting and tracking cell instances with cosine embeddings and recurrent hourglass networks

Christian Payer[a], Darko Štern[b], Marlies Feiner[c], Horst Bischof[a], Martin Urschler[b,d,*]

[a] Institute of Computer Graphics and Vision, Graz University of Technology, Graz, Austria
[b] Ludwig Boltzmann Institute for Clinical Forensic Imaging, Graz, Austria
[c] Division of Phoniatrics, Medical University Graz, Graz, Austria
[d] Department of Computer Science, The University of Auckland, New Zealand

ARTICLE INFO

Article history:
Received 15 February 2019
Revised 5 June 2019
Accepted 26 June 2019
Available online 29 June 2019

Keywords:
Cell
Tracking
Segmentation
Instances
Recurrent
Video
Embeddings

ABSTRACT

Differently to semantic segmentation, instance segmentation assigns unique labels to each individual instance of the same object class. In this work, we propose a novel recurrent fully convolutional network architecture for tracking such instance segmentations over time, which is highly relevant, e.g., in biomedical applications involving cell growth and migration. Our network architecture incorporates convolutional gated recurrent units (ConvGRU) into a stacked hourglass network to utilize temporal information, e.g., from microscopy videos. Moreover, we train our network with a novel embedding loss based on cosine similarities, such that the network predicts unique embeddings for every instance throughout videos, even in the presence of dynamic structural changes due to mitosis of cells. To create the final tracked instance segmentations, the pixel-wise embeddings are clustered among subsequent video frames by using the mean shift algorithm. After showing the performance of the instance segmentation on a static in-house dataset of muscle fibers from H&E-stained microscopy images, we also evaluate our proposed recurrent stacked hourglass network regarding instance segmentation and tracking performance on six datasets from the ISBI celltracking challenge, where it delivers state-of-the-art results.

1. Introduction

Tracking segmented instances of objects throughout videos plays a crucial role in biomedical imaging tasks like analyzing cell growth or migration (Ulman et al., 2017). Ongoing technological improvements in phase enhancing and fluorescence microscopy lead to datasets that show biological information at increasingly higher spatial and temporal resolutions. While this is beneficial for answering to relevant challenges in biomedical research like understanding cell motility and cell proliferation (Zimmer et al., 2006) in the context of biological processes (e.g., immune response (Evans et al., 2009), embryonic development (Montell, 2008), or tumorigenesis (Condeelis and Pollard, 2006)), the growing amount of imaging data also poses challenges for the required data analysis. Manually segmenting and tracking cell instances from huge live-cell datasets is too laborious, thus automatic medical image analysis approaches are required, which have to cope with low signal-to-noise ratio, high cell density, or image artifacts, e.g., from inhomogeneous staining. Moreover, automatic methods have to deal with mitotic and apoptotic events that affect cell lineages as cells move through their environment, as well as large variability in visual appearance, size and number of cells. Recently, predominantly machine learning techniques, e.g., deep convolutional neural networks (LeCun et al., 2015), have been applied to the problem of automatic instance segmentation and tracking, due to their capability of learning from properly annotated datasets how to overcome these challenges.

Differently to semantic segmentation (e.g., (Ronneberger et al., 2015; Payer et al., 2018a)), instance segmentation does not only assign a class label to each pixel of an image, but also distinguishes between instances within each class, e.g., each individual cell or cell nucleus from a microscopy image gets assigned a unique ID. With the high performance of recent fully convolutional neural network architectures like U-Net (Ronneberger et al., 2015), semantic segmentation has also been successfully applied to instance segmentation tasks. However, this approach suffers from

* Corresponding author at: Ludwig Boltzmann Institute for Clinical Forensic Imaging, Graz, Austria
E-mail address: martin.urschler@cfi.lbg.ac.at (M. Urschler).

https://doi.org/10.1016/j.media.2019.06.015

drawbacks due to instances being treated as connected components in postprocessing and for adjoining instances to be separated, artificial border background pixels in between neighboring instances have to be introduced in the semantic segmentation loss.

In computer vision and scene understanding, e.g., to track individual persons on surveillance videos, the need for instance segmentation has led to methods that sequentially segment one instance at a time. In (He et al., 2017), all instances are first detected and independently segmented, while in (Ren and Zemel, 2017), recurrent networks are used to memorize which instances were already segmented. Segmenting one instance at a time can be problematic when hundreds of instances are visible in the image, as often is the case with e.g., cell instance segmentation. More recently proposed methods perform simultaneous segmentation of instances by predicting embeddings for all pixels at once (Newell et al., 2017; Kong and Fowlkes, 2018). Such embeddings have similar values for pixels within the same instance, but differ between instances.

To enable tracking of segmented instances, temporal coherence between frames is an important cue to preserve instance IDs throughout videos. By combining a novel embedding representation with a fully convolutional recurrent neural network, we have proposed an original method for tracking instance segmentations in our MICCAI work (Payer et al., 2018b). There we have shown how to effectively use temporal information in the form of a convolutional gated recurrent unit (Ballas et al., 2016), implemented in a neural network architecture based on the stacked hourglass network (Newell et al., 2016). In this manuscript, we extend our preliminary work (Payer et al., 2018b) by improving and in more detail describing the components of our method, more comprehensive exploration of the related work, as well as more extensive experimental evaluation.

1.1. Related work

A critical step in many biomedical image analysis tasks is the automatic detection and segmentation of cells, cell nuclei, muscle fibers or other sub-millimeter structures from images or videos. This task is challenging due to biological variability, constraints on achievable signal-to-noise ratio and/or the occurrence of artifacts during acquisition. Earlier works addressing these challenges were based on handcrafted features derived from image (pre-)processing methods like thresholding, morphological operations, deformable models (Bergeest and Rohr, 2012), or watershed segmentation either based on edge detection (Wählby et al., 2004) or controlled by markers (Carpenter et al., 2006; Veta et al., 2013). Comprehensive reviews on these approaches can be found in Gurcan et al. (2009); Meijering (2012), and Irshad et al. (2014). Later, the introduction of supervised machine learning has brought a paradigm shift in cell detection and segmentation, as e.g., demonstrated by the excellent performance of the deep learning based cell mitosis detection method from Ciresan et al. (2013) or by recent results on colon gland segmentation (Sirinukunwattana et al., 2017).

Nowadays, deep learning based methods for cell detection and segmentation use convolutional neural networks (CNNs), originally proposed in LeCun et al. (1998). Within this paradigm, earlier methods relied on formulating detection or segmentation as a pixel-wise classification problem and regularizing the pixel-wise predictions in a post-processing step (e.g. (Song et al., 2017; Kainz et al., 2017)). More recently, state-of-the-art performance is achieved using fully convolutional architectures (FCN) (Shelhamer et al., 2017), e.g., the very popular U-Net of Ronneberger et al. (2015) that extends the downsampling and upsampling paths of the FCN by connecting intermediate down- and upsampling levels, thus making use of context information during upsampling. Several segmentation and detection algorithms have been proposed based on variants of this paradigm, for medical images in general (Payer et al., 2019), or more specifically for microscopy images (Kraus et al., 2016; Akram et al., 2016; Li et al., 2018; Xie et al., 2018; Raza et al., 2019).

Extending the problem of semantic segmentation to instance segmentation, where each instance of a segmented object receives its own unique ID, has received a lot of interest in the computer vision literature. Romera-Paredes and Torr (2016) as well as Ren and Zemel (2017) segment each instance individually, with recurrent neural networks memorizing which instances were already segmented. Segmenting solely one instance at a time can be problematic when hundreds of instances are visible in the image. Building upon R-CNN (Girshick et al., 2014), in the Mask R-CNN of He et al. (2017) all instances are first detected and independently segmented. Following up on the original U-Net method (Ronneberger et al., 2015), in medical image analysis applications, all instances are segmented simultaneously by performing a foreground/background segmentation and a connected component analysis as a postprocessing step. These methods have to strongly focus on the borders that separate instances, often by introducing artificial borders in the foreground segmentation groundtruth and an additional loss on these borders (Chen et al., 2017; Xu et al., 2017; Graham et al., 2019). Recent methods in the computer vision community for simultaneously segmenting all instances predict the individual instances directly by encoding them as pixel-wise embeddings (Newell et al., 2017; Kong and Fowlkes, 2018). These embeddings have similar values within a segmented instance, but differ among instances, which is ensured by an embedding loss penalizing Euclidean distance in embedding space (Harley et al., 2017). Differently to methods performing semantic foreground/background segmentations, these methods do not require border enhancement and connected component analysis in the postprocessing step.

In many biological and histopathological applications it is not only important to extract object instances, but also to track them over time. Here, the temporal information is an important cue to establish coherence between frames, thus preserving instances throughout videos, even in the presence of cell mitosis events. The task of cell instance segmentation and tracking has received attention in the form of public challenges like the ISBI celltracking challenge (Maška et al., 2014; Ulman et al., 2017). Participants of such challenges use different approaches like scoring functions for dynamic programming (Magnusson and Jalden, 2012), matching elliptical shapes from frame to frame (Türetken et al., 2017), graph cut (Bensch and Ronneberger, 2015) or probabilistic graphical models (Schiegg et al., 2015; Arbelle et al., 2018) for joint segmentation and tracking, or probabilistic models based on moral lineage tracing on cell detections from an FCN (Rempfler et al., 2018).

To incorporate temporal information into the deep learning framework, recurrent neural networks like the long short-term memory model (LSTM) proposed by Hochreiter and Schmidhuber (1997) or gated recurrent units (GRU) from Cho et al. (2014) have been proposed. Recently, convolutional variants of recurrent neural networks were proposed, in various domains like weather forecasting (Xingjian et al., 2015) or action recognition (Ballas et al., 2016). For incorporating temporal information into FCNs, Tao et al. (2017) proposed to use convolutional LSTMs to increase details for video super resolution, while Tokmakov et al. (2017) use convolutional GRUs (ConvGRUs) to segment moving objects in unconstrained videos. Up to our knowledge, in our preliminary work (Payer et al., 2018b) we were the first to combine a fully convolutional method based on recurrent neural networks with an embedding loss in the context of instance segmentation and tracking.

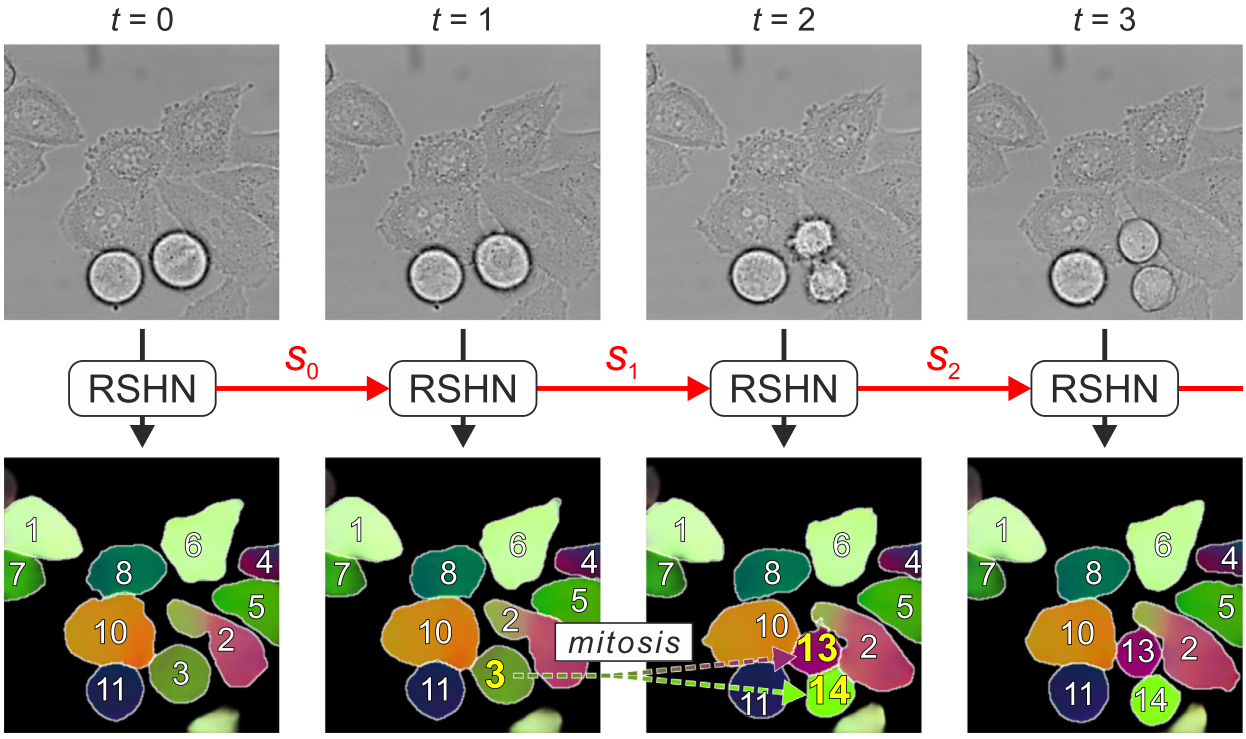


Fig. 1. Overview of our proposed instance segmentation and tracking framework. Our novel recurrent stacked hourglass network (RSHN) generates real valued embedding vectors for extracting cell instances. For every input frame, the internal state $\mathbf{s}_t$ of the RSHN is updated for the next frame in order to propagate embedding vectors and detect mitosis events. The images in the top row show input frames, while the images in the bottom row show three randomly chosen dimensions of the predicted embedding vectors as RGB channels, as well as the detected instances encoded by numbers and outlined with a white border. Cell instance IDs involved in mitosis are visualized in yellow ($\{3\} \rightarrow \{13, 14\}$). (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

1.2. Contributions

In line with our preliminary work in Payer et al. (2018b), in this manuscript we contribute a recurrent fully convolutional network for embedding-based instance segmentation and tracking. To memorize temporal information, we integrate ConvGRUs (Ballas et al., 2016) into a stacked hourglass network (Newell et al., 2016). Furthermore, we use a novel embedding loss based on cosine similarities that requires only neighboring instances to have different embeddings. In addition to these contributions, this manuscript extends our preliminary work in the following ways:

- We overcome the need to downsample high resolution images by using a tiling strategy that now allows us to work on arbitrary size images.
- We replaced the previous algorithm for clustering embeddings with the mean shift approach, which solely requires a single tuning parameter.
- We provide a more in-depth description of our proposed method and its implementation details.
- We evaluated our proposed method on an additional in-house dataset for muscle fiber segmentation, where our cosine embedding loss shows improvements as compared to a classical softmax cross entropy loss.
- We considerably improved our preliminary results on the six datasets of the ISBI celltracking challenge, which show large variability in visual appearance, size and number of cells, thus demonstrating the wide applicability of our approach.

2. Instance segmentation and tracking

Fig. 1 shows our proposed framework for instance segmentation and tracking. To distinguish instances, they are represented as real valued embedding vectors for each pixel at different time points. By representing temporal sequences of embeddings in a recurrent neural network, a predictor can be learned from the data, which allows tracking of embeddings also in the case of events involving dynamic structural changes, e.g., cell mitosis events. To finally generate instance segmentations and assign a unique ID for each instance, we perform spatiotemporal clustering of the predicted embeddings implemented with a tiling strategy for overlapping image regions and consecutive frame pairs.

2.1. Recurrent stacked hourglass network

We use the stacked hourglass network (Newell et al., 2016) as a basis for our recurrent network architecture. The hourglass network is similar to the U-Net (Ronneberger et al., 2015), i.e., it consists of convolution layers in a contracting and an expanding path for multiple levels, but it additionally introduces convolution layers in the split connections as a parallel path. We exchange these convolution layers in the parallel path by ConvGRUs (Ballas et al., 2016), which allow us to propagate temporal video information within a fully convolutional network architecture. A schematic overview of this recurrent stacked hourglass network (RSHN) is shown in Fig. 2. Each of the ConvGRUs has its own internal state $\mathbf{s}_t$ at timestep t, which has a size that is equal to the size of its input. For $t = 0$ the state $\mathbf{s}_t$ is initialized with zeroes. Based on its current input and current value, the internal state $\mathbf{s}_t$ is updated to the new state $\mathbf{s}_{t+1}$ after each timestep. Thus, by consecutively providing the RSHN with individual frames, information from previous frames is encoded in the current state and propagated to the next frame. For more information on the state update equations, we refer to Ballas et al. (2016).

Differently from the original stacked hourglass network, we use single convolution layers with 3×3 filters and 64 outputs for all

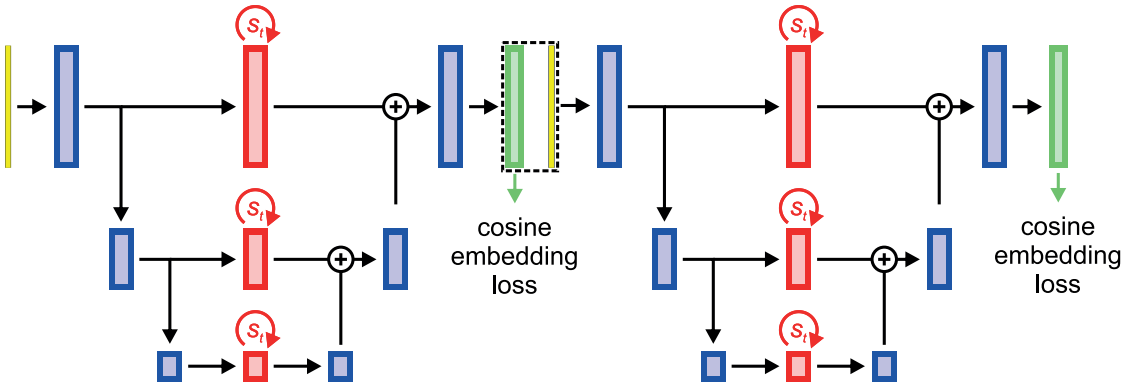
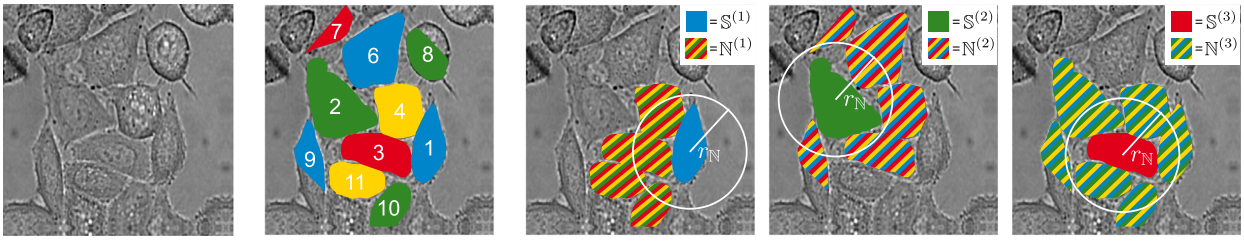


Fig. 2. Schematic overview of our proposed recurrent stacked hourglass network (RSHN) with two hourglasses and three levels. In our implementation, we use recurrent stacked hourglass networks with seven levels. Yellow bars: input; blue boxes: convolutions; red boxes: ConvGRU; dashed black box: concatenation; green boxes: embeddings. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)



(a) Input image. (b) Colored instances. (c) Individual instances, with differently colored neighbors.

Fig. 3. Visualization for coloring neighboring instances. In theory, four colors are sufficient to find a coloring of neighboring instances, such that no neighboring instances have the same color. The first image shows the input cells; the second image shows the four-colored groundtruth instances. The last three images show colored example instances and their neighbors. As long as the neighboring instances have a different color (e.g., for the instance with ID 1: $\mathbb{S}^{(1)}$ is blue, $\mathbb{N}^{(1)}$ is either yellow, green, or red), the instance is easily distinguishable. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

blocks in the contracting and expanding paths. To represent states, we use ConvGRU with 3×3 filters and 64 outputs in between paths to represent states. We use max pooling with kernel size of 2×2 as downsampling in the contracting path and nearest neighbor upsampling in the expanding path. For generating the next lower or upper level, we use 2 as the down- and upsampling factor. We use addition to combine the outputs of the parallel path and the expanding path at the corresponding levels, see Fig. 2.

As proposed by (Newell et al., 2016), we stack two hourglasses in a row to improve network predictions, since the second hourglass refines possibly erroneous predictions of the first one. Therefore, we concatenate the output of the first hourglass with the input image to use it as input for the second hourglass. We apply the loss function on the outputs of both hourglasses during training, while we only use the outputs of the second hourglass during network inference and the clustering of embeddings.

2.2. Cosine embedding loss

We let our RSHN predict a d-dimensional embedding vector $\mathbf{e}_p \in \mathbb{R}^d$ for each pixel p of an image. The embedding vectors need to have the following two properties to allow the separation of all instances $i \in \mathbb{I}$. Firstly, embeddings of pixels $p \in \mathbb{S}^{(i)}$ belonging to the same instance i need to be similar. Secondly, embeddings of $\mathbb{S}^{(i)}$ need to be dissimilar to embeddings of pixels $p \in \mathbb{S}^{(j)}$ of other instances $j \in \mathbb{I}$ with $j \neq i$.

Following from the four color map theorem (Appel and Haken, 1976), only neighboring instances need to have different embeddings in order to distinguish between them, while instances that are not direct neighbors may get assigned the same embedding, since they are spatially distinguishable. For our proposed embedding loss, we relax the need of dissimilarity between all different instances only to the neighboring ones, i.e., $\mathbb{N}^{(i)} = \bigcup_j \mathbb{S}^{(j)}$ for all instances $j \neq i$, where the distance of any pixel $p \in \mathbb{S}^{(j)}$ to the center of gravity of all pixels $p \in \mathbb{S}^{(i)}$ is less than $r_{\mathbb{N}}$. This relaxation simplifies the problem by assigning only a limited number of different embeddings to a possibly large number of different instances. Fig. 3 visualizes $\mathbb{N}^{(i)}$ and $\mathbb{S}^{(i)}$ for instances on an example image.

For video sequences, we define the pixels $p \in \mathbb{S}^{(i)}$ and $p \in \mathbb{N}^{(i)}$ not only for single frames, but for a sequence of frames with length l. This way, if the two properties of the embeddings are satisfied over a sequence of frames, the embeddings of the same instance stay similar over the sequence of frames, while the embeddings of neighboring instances are dissimilar. This also holds for instances involved in, e.g., mitosis events, which are neighbors in time.

We compare two embeddings $\mathbf{e}_1, \mathbf{e}_2$ with the cosine similarity

$$\cos(\mathbf{e}_1, \mathbf{e}_2) = \frac{\mathbf{e}_1 \cdot \mathbf{e}_2}{\|\mathbf{e}_1\| \|\mathbf{e}_2\|}, \tag{1}$$

which ranges from -1 to 1, where -1 indicates the vectors have the opposite, 0 orthogonal, and 1 the same direction.

We define the cosine embedding loss for each individual instance $i \in \mathbb{I}$ as

$$L^{(i)} = L^{(i)}_{\mathbb{S}} + L^{(i)}_{\mathbb{N}}. \tag{2}$$

Here, the first term $L^{(i)}_{\mathbb{S}}$ is defined as

$$L^{(i)}_{\mathbb{S}} = 1 - \frac{1}{|\mathbb{S}^{(i)}|} \sum_{p \in \mathbb{S}^{(i)}} \cos(\bar{\mathbf{e}}^{(i)}, \mathbf{e}_p)^2, \tag{3}$$

while the second term $L^{(i)}_{\mathbb{N}}$ is defined as

$$L^{(i)}_{\mathbb{N}} = \frac{1}{|\mathbb{N}^{(i)}|} \sum_{p \in \mathbb{N}^{(i)}} \cos(\bar{\mathbf{e}}^{(i)}, \mathbf{e}_p)^2, \tag{4}$$

with the mean embedding of instance i being defined as

$$\bar{\mathbf{e}}^{(i)} = \frac{1}{|\mathbb{S}^{(i)}|} \sum_{p \in \mathbb{S}^{(i)}} \mathbf{e}_p. \tag{5}$$

By minimizing $L^{(i)}$, the first term $L_{\mathbb{S}}^{(i)}$ urges embeddings $\mathbf{e}_p$ of pixels $p \in \mathbb{S}^{(i)}$ to have the same direction as the mean embedding $\bar{\mathbf{e}}^{(i)}$, which is the case when $\cos(\bar{\mathbf{e}}^{(i)}, \mathbf{e}_p) \approx 1$. Thus, this term favors solutions, where the variation of embeddings $\mathbf{e}_p$ of pixels $p \in \mathbb{S}^{(i)}$ is small as compared to the mean embedding $\bar{\mathbf{e}}^{(i)}$. The second term $L_{\mathbb{N}}^{(i)}$ urges embeddings $\mathbf{e}_p$ of pixels $p \in \mathbb{N}^{(i)}$ to be orthogonal to the mean embedding $\bar{\mathbf{e}}^{(i)}$, i.e., $\cos(\bar{\mathbf{e}}^{(i)}, \mathbf{e}_p) \approx 0$. Differently to our preliminary work (Payer et al., 2018b), we now not only use the squared cosine similarity in $L_{\mathbb{N}}^{(i)}$, but also in $L_{\mathbb{S}}^{(i)}$, which we found to result in smoother embeddings within instances and in faster network convergence.

In order to distinguish between background and foreground instances $i \in \mathbb{I}$, we treat the background instance b with the same cosine embedding loss as defined in (2). For b we define $\mathbb{S}^{(b)}$ to contain all background pixels and $\mathbb{N}^{(b)}$ to contain all other pixels, i.e., pixels of all annotated instances. Thus, the embeddings of all pixels of the background are similar, while embeddings of cells are dissimilar to the embedding of the background. To more easily identify the background region for the final clustering of the embeddings, we do not calculate the mean embedding $\bar{\mathbf{e}}^{(b)}$ for the background b, but set it fixed to be a d-dimensional vector with the entry 1 in the first dimension and 0 in all the other dimensions, i.e., $\bar{\mathbf{e}}^{(b)} = (1, 0, \ldots, 0)$.

The final loss for each image is defined as the sum of the loss of the background b and the accumulated losses for the foreground instances $i \in \mathbb{I}$, i.e.,

$$L = L^{(b)} + \frac{1}{|\mathbb{I}|} \sum_{i \in \mathbb{I}} L^{(i)}. \tag{6}$$

Thus, due to all foreground instances $i \in \mathbb{I}$ being neighboring instances of the background instance b, minimizing this loss urges all foreground instances to generate embeddings that are different to the fixed background embedding $\bar{\mathbf{e}}^{(b)}$.

2.3. Clustering embeddings into instances

To get the final segmentations from the predicted pixel-wise embeddings, we need to group them into individual instances. As we know the target embedding vector $\bar{\mathbf{e}}^{(b)}$ of the background pixels, we identify the background region b as pixels p with $\cos(\bar{\mathbf{e}}^{(b)}, \mathbf{e}_p)^2 > 0.5$. For the remaining foreground pixels, i.e., pixels of the individual instances $i \in \mathbb{I}$, the exact values of the mean embedding vectors $\bar{\mathbf{e}}^{(i)}$ are unknown. However, due to our proposed embedding loss function, we know that for each instance i the cosine similarity $\cos(\mathbf{e}_p, \bar{\mathbf{e}}^{(i)})^2$ for pixels $p \in \mathbb{S}^{(i)}$ is ≈ 0, and for pixels $p \in \mathbb{N}^{(i)}$ it is ≈ 1. Therefore, we determine the individual instances by identifying cluster centers in the space of the embedding vectors. As the number of instances on an image is not known in advance, we need to perform this clustering with a method that is able to identify the number of clusters on its own. Differently to our preliminary work (Payer et al., 2018b), we do not use HDBSCAN (Campello et al., 2015), but the mean shift (Comaniciu and Meer, 2002) algorithm, modified to use cosine similarities. The benefit of mean shift compared to HDBSCAN is that we only need to set a single bandwidth parameter h, which is connected to the cluster distance in embedding space.

As in the d-dimensional embedding space the neighborhood of pixels in the spatial domain is lost, we append the x and y coordinates of p multiplied with a factor c to each embedding vector $\mathbf{e}_p$. This way, the modified $d+2$-dimensional embedding vectors also incorporate spatial information, which we found to reduce noisy predictions on the borders of cells. Furthermore, we remove instances that have an area of less than t_{size}, which reduces wrong predictions due to image noise, e.g., dust or speckles.

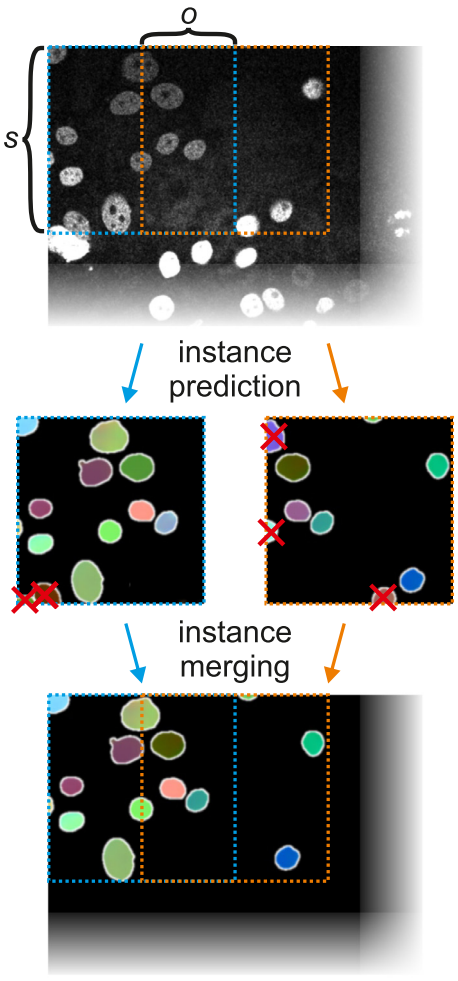


Fig. 4. Visualization of the merging strategy for predicting instances in overlapping tiles. Instances touching the border of a tile are removed, as long as the border is not an outer border of the whole image. If instances are predicted by multiple tiles, the instances with the largest area are taken.

2.4. Predicting instances for variably sized images

As our framework allows predicting of instances on images with variable size, as well as videos of variable length, we employ a tiling strategy for overlapping images or image sequences. Extending our preliminary work from Payer et al. (2018b), our proposed framework allows training and testing on images with arbitrary pixel resolution, by employing a tiling strategy similar to Ronneberger et al. (2015). When training the networks, we randomly crop a tile of size s from the training image, while during inference, we split the image into multiple tiles of size s that overlap by o pixels in x and y direction. For each tile the instances are predicted as described in Section 2.3. Due to our tiling strategy, images no longer need to be heavily downsampled to fit the network into the limited GPU memory as it was the case in Payer et al. (2018b).

To predict instances of images with arbitrary pixel resolution, we use the following merging strategy for overlapping tiles, as visualized in Fig. 4: Except for instances that are touching an outer border of the image, we remove all instances that are touching a tile border, as these instances are more uncertain due to the re-

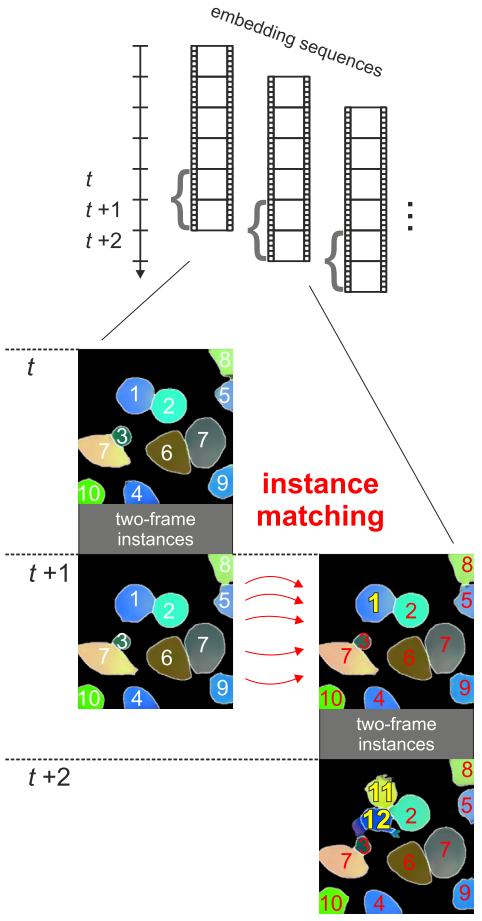


Fig. 5. Visualization of merging instances for overlapping frame pairs. Embeddings are predicted for a sequence of frames with length l, while the embeddings of the last two frames of this sequence are used to generate two-frame instances. Matching instances in the overlapping frame $t+1$ are visualized with red IDs, while instances involved in mitosis events ($\{1\} \rightarrow \{11, 12\}$) are visualized in yellow. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

duced spatial context. Moreover, for the neighboring overlapping tile, the predictions for the same instances are more likely to be located in the center of the tile, instead of touching its border. If after removal, for an instance there are still several instance candidates predicted by multiple overlapping tiles, we select the instance candidate with the largest area.

2.5. Propagating instances over time

Our framework allows predicting instances for videos of variable length, by separating videos into shorter overlapping sequences of length l shifted by one frame, and by matching instances on two corresponding frames from consecutive overlapping sequences, see Fig. 5.

Due to our loss function incorporating instances over an image sequence of length l, the network is trained to predict the same embedding for the same instance throughout any sequence of length l. Although the ConvGRU in our RSHN in principle allows predicting videos of arbitrary length, for long sequences, instances with same embeddings that are not neighboring in the beginning may become neighbors at later frames due to cell movements. As this would violate our assumption that neighboring instances have dissimilar embeddings, we predict embeddings for sequences that have the same length l as sequences used for training the networks. At the beginning of all predicted sequences, i.e., when predicting the first frame of the sequence, we initialize the states of the ConvGRU with zeroes, such that all instances may get assigned new embeddings.

As it is not allowed for an instance to disappear and reoccur later in time in our evaluated datasets, it is sufficient to identify every instance for consecutive frame pairs. Therefore, we do not cluster the embeddings for the whole image sequence, but only for the last two consecutive frames, resulting in two-frame instances, with consistent IDs for same instances and different IDs assigned for instances involved in splits due to mitosis events. Starting from the first sequence, we predict instances for its last two frames $\langle t, t+1 \rangle$. Then, we match these instances with instances predicted by the overlapping frame pair $\langle t+1, t+2 \rangle$ from the next sequence, which was shifted by one frame. To achieve this, we identify same instances in the overlapping frame $t+1$ of the frame pairs $\langle t, t+1 \rangle$ and $\langle t+1, t+2 \rangle$ by the highest intersection over union (IoU) between all segmented instances in the overlapping frame. Instances that only exist in the second frame $t+2$ of the matched overlapping frame pair, i.e., instances created due to mitosis, are assigned new IDs. As a special case, to segment instances for the first $l-2$ frames of the whole video, we predict sequences of length $2, 3, \ldots l-1$ and merge the instances with the same matching strategy. Due to noise in the microscopy images, we additionally remove instances that exist for less than t_{length} frames and correct wrongly detected mitosis events, i.e., instances that split and merge again within t_{merge} frames.

3. Experimental setup

3.1. Training and implementation details

We train the networks with TensorFlow[1] and perform on-the-fly data augmentation using SimpleITK[2]. For reproducing our results for the ISBI celltracking challenge, the trained network models and code used for generating the results are available on the challenge's website[3]. Additionally, we made the source code of our training framework publicly available[4]. We determined the training hyperparameters with initial cross-validation experiments on the datasets. We set the network parameters of the RSHN as follows: We use hourglass networks with seven levels and an input size of $s = 256 \times 256$. The weights of each convolution layer are initialized with the method as described in (He et al., 2015), the biases with 0. The convolution layers, as well as ConvGRU, use ReLU (Glorot et al., 2011) as an activation function. The networks do not employ any normalization layers or dropout, but use an L2 weight regularization factor of 0.00001. Due to the demanding training of recurrent neural networks, in terms of both memory and computational requirements, we set the mini-batch size to 1. We train the recurrent networks for sequences of $l = 8$ consecutive frames and for predicting embeddings of size 16. We train all networks with ADAM (Kingma and Ba, 2015) for total 60,000 iterations, while we start with a learning rate of 0.0001, and reduce it to 0.00001 after 30,000 iterations. Training of a recurrent network takes $\approx$7 hours on a single NVIDIA Titan V with 12 GB. Network inference with subsequent instance prediction takes from 4.5s per frame (dataset DIC-C2DH-HeLa, $\approx$25 instances per frame, no overlapping tiles) to 140s per frame (dataset Fluo-N2DL-HeLa, $\approx$500 instances per frame, 40 overlapping tiles). For the clustering with mean shift, we set $h = 0.1$ and $c = 0.005$.

For training data augmentation, we change input intensity values and perform spatial deformations. First, due to image noise,

[1] https://www.tensorflow.org/.
[2] http://www.simpleitk.org/.
[3] http://celltrackingchallenge.net/participants/TUG-AT/.
[4] https://github.com/christianpayer/.

we perform Gaussian smoothing with $\sigma = 1$ for every input image. Then, we change the image intensity values such that the minimum and maximum values are approximately -1 and 1. As microscopy images may contain outliers in terms of minimum and maximum intensity values, we calculate the robust minimum v_{min} as the intensity value that is larger than 0.1% of all intensity values of an image, and the robust maximum v_{max} as the value that is larger than 99.9% of all intensity values. Then, for data augmentation, we randomly sample values from a uniform distribution within the following specified intervals. We shift each intensity value randomly by $v_{\text{shift}} = [-0.65, 0.65]$ and scale each intensity by $v_{\text{scale}} = [0.35, 1.65]$. As our proposed framework splits the input image into multiple tiles for inference, for training, we randomly crop a region of the desired network size from the input image. For the random spatial deformations in both x and y axes, we mirror along the axis with probability $f_p = 0.5$, rotate by $r = [-180^\circ, 180^\circ]$ and scale by $sc = [0.5, 1.5]$. Furthermore, we employ elastic deformations, by randomly moving points by $b = [-10, 10]$ pixels on a grid of size $g = 6 \times 6$ and interpolating with third order splines.

3.2. Evaluation metrics

As evaluation metrics, we use the measures of the ISBI celltracking challenge (Ulman et al., 2017), which represent segmentation (SEG), tracking (TRA) and overall performance (OP). In the SEG measure, a groundtruth instance and a predicted instance are considered matching, if and only if their intersection over union (IoU) is larger than half the size of the groundtruth instance. Note that for each groundtruth instance only one predicted instance may satisfy this condition. For each groundtruth instance, if a match was found, its SEG value is set to the IoU; otherwise, it is set to 0. The final SEG measure is the mean value over all SEG values of all groundtruth instances. In the TRA measure, the groundtruth and predicted cell family trees are compared. This metric counts the changes needed to transform the predicted tree into the groundtruth tree. The OP measure evaluates the performance of both segmentation and tracking combined, and is defined as the mean of the SEG and TRA measures. For more details on the metrics, we refer to Ulman et al. (2017).

4. Datasets

We perform experiments on an in-house muscle fiber dataset for instance segmentation, and on six datasets of the ISBI celltracking challenge for instance segmentation and tracking (Ulman et al., 2017).

4.1. Muscle fiber dataset

We use an in-house muscle fiber dataset involving H&E-stained microscopy images from sheep vocal cord muscles (see Fig. 6) for evaluating cell instance segmentation on still images. The dataset was provided by the Division of Phoniatrics at Medical University of Graz, and was collected in a project involving functional electrical stimulation (Bennie et al., 2002) to reverse aging induced voice changes, i.e., presbyphonia (Karbiener et al., 2016). The dataset consists of ten images with a mean pixel resolution of 9644×7248 pixels, each containing several hundred muscle fiber cell instances. To have a groundtruth for instance segmentation, pixel-wise annotation of the dataset was performed by a medical student under supervision of an expert with more than 10 years experience in histological imaging.

Before training the networks, we resample the images to a size of 1024×1024 pixel. As all our networks work with an input size of $s = 256 \times 256$, we perform the tiling strategy as described in Section 2.4, where the tiles overlap with $o = 128$ pixels.

As this dataset does not contain tracking information, we only evaluate the SEG measure (see Section 3.2).

Table 1
Dataset dependent parameters for the datasets of the ISBI celltracking challenge.

Dataset	Resampled Size	t_{size}	r_{N}
DIC-C2DH-HeLa	256×256	500	50
Fluo-C2DL-MSC	256×256	100	150
Fluo-N2DH-GOWT1	512×512	100	50
Fluo-N2DL-HeLa	1100×700	50	10
PhC-C2DH-U373	512×384	500	50
Fluo-N2DH-SIM+	512×512	100	50

4.2. Cell tracking challenge dataset

We use six different datasets of cell microscopy videos from the ISBI celltracking challenge (Ulman et al., 2017) for evaluating cell instance segmentation and tracking, namely DIC-C2DH-HeLa, Fluo-C2DL-MSC, Fluo-N2DH-GOWT1, Fluo-N2DL-HeLa, PhC-C2DH-U373, and Fluo-N2DH-SIM+ (see Fig. 8). Each celltracking dataset consists of two annotated training videos and two testing videos with image sizes ranging from 512×512 to 1200×1024 and with 48 to 138 frames. We refer to (Maška et al., 2014) for details on imaging and video setup. We did not evaluate our proposed algorithm on the only remaining 2D dataset of the ISBI celltracking challenge, i.e., PhC-C2DL-PSC, due to the small cell sizes and large amount of cell instances up to approximately a thousand, which resulted in an increased runtime that prevented our method to generate results on the celltracking challenge servers in a reasonable amount of time. For some datasets, the original input resolution is very high such that without resampling the images to a lower resolution, there is not enough context information for our network. Therefore, we resample the training and testing videos for each dataset to the sizes shown in Table 1. As all our networks work with an input resolution of $s = 256 \times 256$, we perform the tiling strategy as described in Section 2.4, where the tiles overlap with $o = 128$ pixels. For datasets, where we resampled the input images, we resample the final predicted segmentations back to the original input resolution. As described in Section 2.5, we predict overlapping sequences of $l = 8$ frames and reset the states of the ConvGRU for each overlapping sequence. We use the last two frames of each sequence for merging overlapping frame pairs and to propagate the instances over time (see Fig. 5).

Each dataset of the ISBI celltracking challenge consists of two types of groundtruth annotations, i.e., the segmentation and tracking groundtruth. The segmentation groundtruth is defined only on a small number of frames of the video. For each annotated frame, either all cell instances, or a subset of the visible instances are fully segmented. The tracking groundtruth is defined for all frames of the video and contains a dot annotation inside each cell instance that is consistent throughout the whole video. As the instance IDs in groundtruth images are consistent throughout the whole video only for tracking, but not for segmentation, we merge both tracking and segmentation groundtruth for each frame to have consistent instance IDs. Although for most frames, there exist only the dot annotations, we still can use the dot annotations for calculating our proposed cosine loss, as we can be sure that the dot annotations are fully inside the instances to be segmented. However, for learning the background embeddings, we can only use the frames on which every cell is segmented. For training the networks, we randomly choose a sequence of frames such that approximately

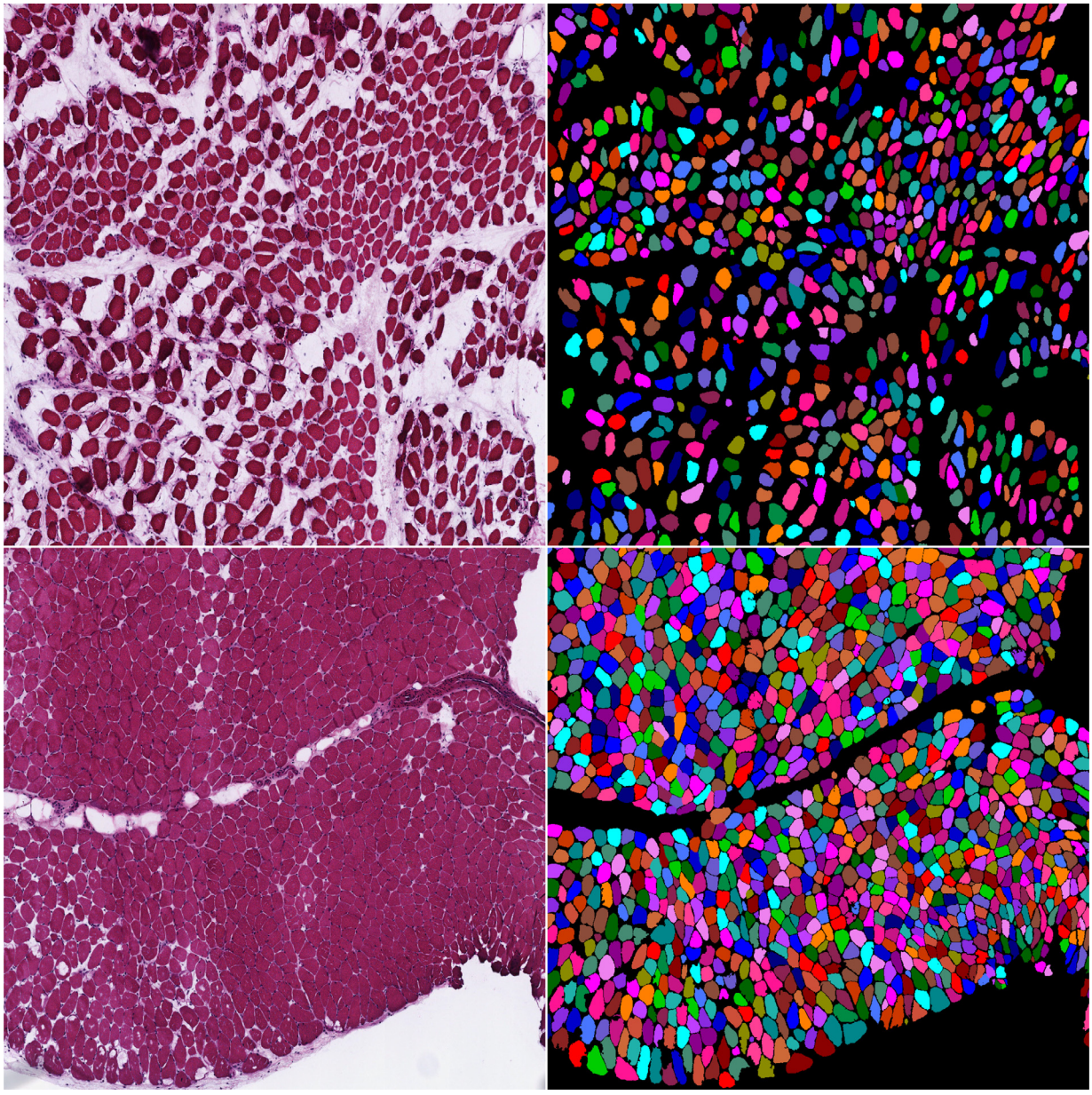


Fig. 6. Example input images and instance segmentation groundtruth for our in-house muscle fiber dataset of H&E-stained microscopy images.

every second mini-batch includes a frame that has a segmentation groundtruth.

The remaining parameters for the datasets are as follows: For removing instances that exist for a too short time period, we set $t_{\text{length}} = 2$. For fixing wrongly detected mitosis events, we set $t_{\text{merge}} = 10$. Parameters that are different for each dataset are shown in Table 1.

As evaluation metrics, we use the SEG and TRA measures, as well as the OP measure which is the mean of the former two (see Section 3.2). The SEG measure is calculated on the predicted instance segmentation images. For the TRA metric, the framework is required to identify the parent ID of each cell. As the framework is able to identify splitting cells and to assign new instance IDs (i.e., mitosis as seen on Fig. 1), the parent ID of each newly created instance is determined as the instance with the highest IoU in previous frames. We further postprocess the cells' family tree to be consistent with the evaluation criteria, e.g., an instance ID may not be used after splitting into children.

5. Evaluation and results

In this paper we proposed a method that performs cell segmentation by representing each cell instance as an embedding vector, as well as incorporates temporal information with recurrent layers.

We evaluate the performance of our novel cosine embedding loss for instance segmentation on the muscle fiber dataset (see Section 4.1) where we compare our loss to the method CellProfiler (Carpenter et al., 2006), which is not learning based, and to the widely used softmax cross entropy loss, which treats all cells as belonging to a single foreground class (see Section 5.1).

To evaluate cell segmentation and tracking performance of our RSHN architecture that incorporates temporal information by integrating ConvGRU layers, we use the ISBI celltracking challenge dataset (see Section 4.2). In Section 5.2, we compare our RSHN architecture with a network architecture that incorporates temporal information by treating the time dimension as an additional spatial dimension, i.e., generating a volumetric input image by

stacking a sequence of images. We also use the same dataset and setup to compare mean shift with different bandwidth parameters h and HDBSCAN for clustering the embedding vectors into cell IDs, see Section 5.3.

Finally, we compare our method with state-of-the-art methods for cell segmentation and tracking by submitting results for six datasets of the open ISBI celltracking challenge, see Section 5.4.

5.1. Evaluation of the cosine embedding loss

To show the performance of the instance segmentation component of our algorithm that is based on the cosine embedding loss, we use the muscle fiber dataset evaluated with three fold cross validation of distinct sets. Since in this experiment there is no temporal information, we use a non-recurrent version of our stacked hourglass network, where each ConvGRU in the parallel path is replaced by a single convolution layer. Using the same network architecture, we compare our proposed cosine embedding loss for instance segmentation with a weighted softmax cross-entropy loss for semantic foreground/background segmentation. As for this semantic segmentation loss, each cell instance must be separated, we use the same border enhancement strategy as proposed by U-Net (Ronneberger et al., 2015), which introduces artificial borders with high loss weight in between touching instances. We generate artificial borders in the groundtruth and set parameters regarding border weights according to the original U-Net paper.

To compare to methods that are not machine learning based, we additionally create instances with CellProfiler (Carpenter et al., 2006), which is a commonly used tool for cell instance segmentation. We carefully tested and combined different filters and modules of CellProfiler and manually tuned various parameters. The final instances were generated from a set of parameters that we found to deliver the best results by visual inspection.

From the results in Table 2, we can see that the learning based methods deliver better results as compared to CellProfiler, while our proposed embedding loss performs better as compared to the softmax cross entropy loss. Specifically, starting from a baseline in the SEG metric of 0.780 for CellProfiler, the results improve to 0.886 for softmax cross entropy, while the networks trained with our proposed cosine embedding loss further increase the metric to 0.911. Thus, learning based methods drastically improve results without requiring carefully tuned parameters. Furthermore, without the need of introducing artificial borders in between cells and/or adapting the loss to focus more on these artificial borders, as often is the case for semantic foreground/background segmentation (Ronneberger et al., 2015; Chen et al., 2017; Xu et al., 2017; Graham et al., 2019), our proposed loss has better performance. This is especially the case for dense and touching cells, where instances are not clearly separated, as shown in Fig. 7.

Table 2
Results of the three fold cross validation experiment comparing CellProfiler that uses image processing, networks trained with our proposed cosine embedding loss for instance segmentation, and networks trained with the weighted softmax cross entropy loss for foreground/background segmentation. Values show mean ± stddev. of the SEG measure for all annotated instances of all ten images.

Method	SEG
Cosine Embedding	**0.911** ± 0.148
Softmax Cross Entropy	0.886 ± 0.152
CellProfiler (Carpenter et al., 2006)	0.780 ± 0.245

5.2. Evaluation of integrating temporal information

In this experiment, we compare our RSHN architecture with a network architecture that incorporates temporal information by treating the time dimension as an additional spatial dimension, i.e., generating a volumetric input image by stacking a sequence of images.

In order to compare different approaches of integrating time information into convolutional neural networks, we compare our proposed RSHN utilizing ConvGRU with a stacked hourglass network that incorporates temporal information by treating the time dimension as an additional spatial dimension of the input image. Thus, the RSHN is compared to the network that does not propagate temporal information via an internal state, but uses a volumetric image as input that is generated by stacking the sequence of 2D images. To obtain this network, we replaced all 2D convolution layers having kernel size 3×3 with 3D convolution layers having kernel size $3 \times 3 \times 3$. Additionally, we replaced the ConvGRU with a convolution layer having kernel size $3 \times 3 \times 3$. The networks are trained on sequences of $l = 8$ frames, while we use our proposed loss function and tiling strategy as described in Section 2. Runtime and memory consumption are approximately the same for both the network with volumetric kernels and the RSHN.

We perform a two fold cross validation experiment for three datasets of the ISBI celltracking challenge (DIC-C2DH-HeLa, Fluo-N2DH-GOWT1, and PhC-C2DH-U373), where we train on one video and evaluate on the other one. The remaining three datasets (Fluo-C2DL-MSC, Fluo-N2DL-HeLa, and Fluo-N2DH-SIM+) were omitted from this experiment, since the two training videos per dataset are too different either in terms of field of view (Fluo-C2DL-MSC), cell shape (Fluo-N2DH-SIM+), or number of visible cells (Fluo-N2DL-HeLa).

Results comparing our RSHN architecture with the stacked hourglass network using volumetric kernels (Volumetric) are shown in Table 3. The results of this experiment show that our RSHN integrating time information with ConvGRU performs better in terms of both segmentation and tracking compared to the network that integrates time as an additional, third dimension of input images.

5.3. Evaluation of clustering the embedding vectors

In this experiment, we evaluate the influence of the bandwidth parameter h of mean shift and compare mean shift clustering with the more complex HDBSCAN, which we used in our preliminary work (Payer et al., 2018a). For this experiment, we use the same cross validation setup with the same trained networks (RSHN) and predicted embedding vectors as used for the previous experiment (see Section 5.2) and only exchange the clustering algorithms. When using mean shift, we set its single bandwidth parameter h to be one out of {0.025, 0.05, 0.1, 0.15, 0.2}. When using HDBSCAN, we set it up to use the same parameters as in (Payer et al., 2018a). Results comparing different values for h for mean shift and HDBSCAN are shown in Table 3.

The similar performance for the different values of h indicate that the performance of our algorithm is not sensitive to this parameter. For small values up to $h = 0.15$ the performance is similar, while for $h = 0.2$ neighboring cell clusters start to merge, resulting in a performance drop in the dataset DIC-C2DH-HeLa. Therefore, we set $h = 0.1$ for all further experiments. When comparing mean shift with HDBSCAN, mean shift performs better in the dataset DIC-C2DH-HeLa, however, for the datasets Fluo-N2DH-GOWT1 and PhC-C2DH-U373 both clustering methods produce similar results. Nevertheless, differently from HDBSCAN, which involves two parameters that are dataset dependent, our proposed mean shift has only a single parameter $h = 0.1$, which we set equal for all datasets.

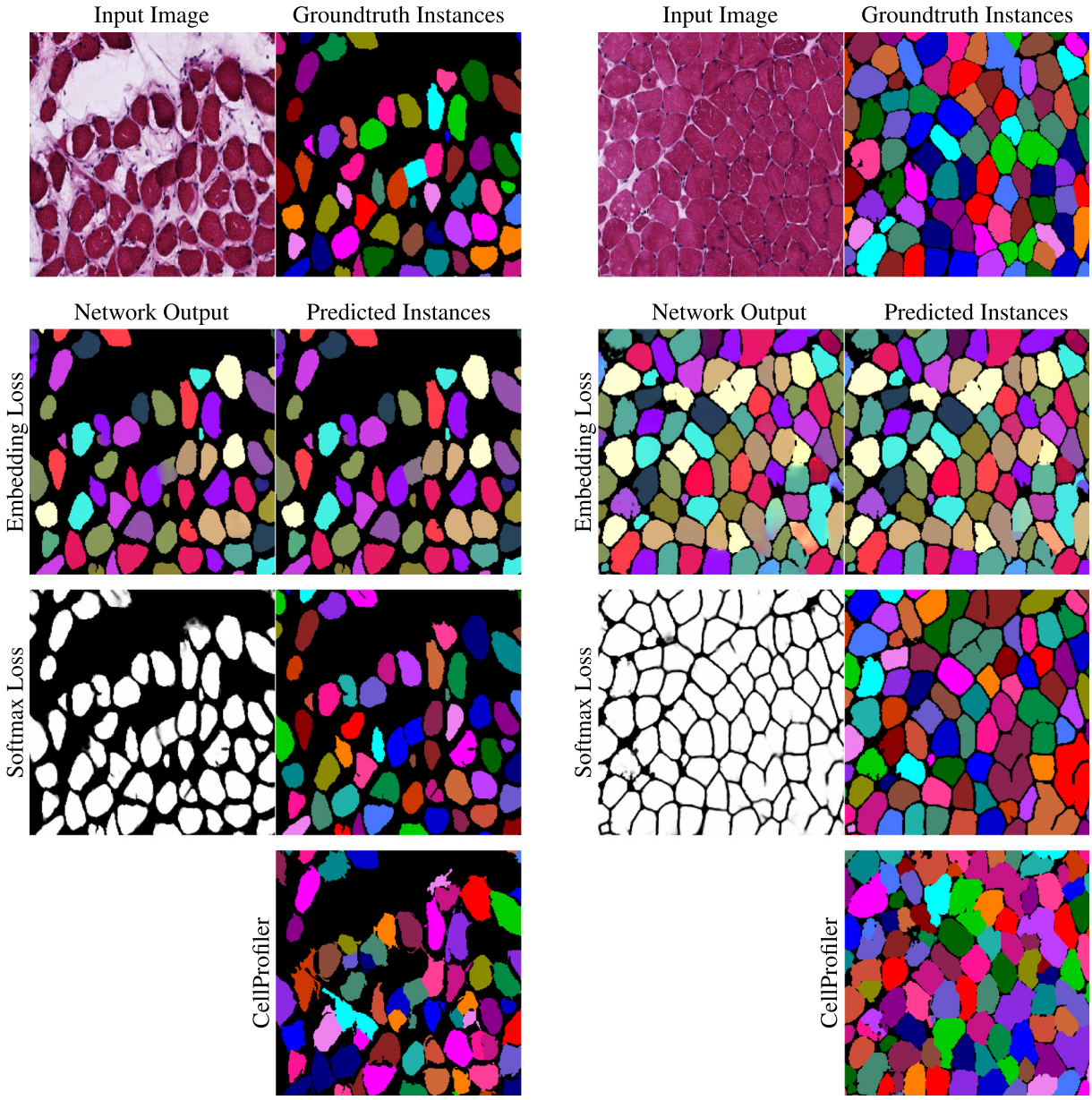


Fig. 7. Examples for two different image tiles of the in-house muscle fiber dataset. The images on the first row show the example input image and groundtruth instance segmentation. For the network outputs with the embedding loss, we show three random embedding dimensions as RGB channels, while for the outputs with the softmax loss, we show the foreground channel. For the embedding loss, the predicted instances are colored with the mean RGB value of the embedding vectors, while for the softmax loss and CellProfiler, they are colored randomly. On the example on the left side, both cosine embedding loss and weighted softmax cross entropy loss deliver good results, while the CellProfiler outputs are noisier. The example on the right side shows results for dense instances. Here, for most cells, our proposed cosine embedding loss is able to detect individual instances, even when the border between cells is not visible. However, especially on the bottom right corner, we can see that the network predicts gradients in some cells, which lead to too many detected instances. The weighted softmax cross entropy loss has problems with detecting the individual instances, although the networks were trained with artificially introduced borders in between touching instances, which are weighted higher in the loss function. CellProfiler also has problems with dense cells, resulting in merged instances with noisy borders.

5.4. Evaluation of cell segmentation and tracking

In our final and main experiment, we compare to other state-of-the-art methods for cell instance segmentation and tracking by evaluating on six datasets of the ISBI celltracking challenge test set, which show large variability in visual appearance, size and number of cells (see Fig. 8). As the groundtruth annotations for the two testing videos of each dataset are only known to the challenge organizers, each challenge participant has to submit the predicted instance segmentation and tracking results, as well as the software executable that has been used to generate these results. The organizers then reproduce the results with the submitted software, perform the evaluations to calculate the individual metrics, and announce the results to the participant, which enables a fair comparison of all submitted methods.

For each of the six evaluated datasets, we trained a single network on both annotated training videos using hyperparameters determined with initial experiments on the training sets. Examples for predicted embeddings and clustered instances are shown in Fig. 8. The results in comparison with the top performing methods of the ISBI celltracking challenge are presented in Table 4.

In the overall performance metric, our method achieves one first and one second rank, showing state-of-the-art results.

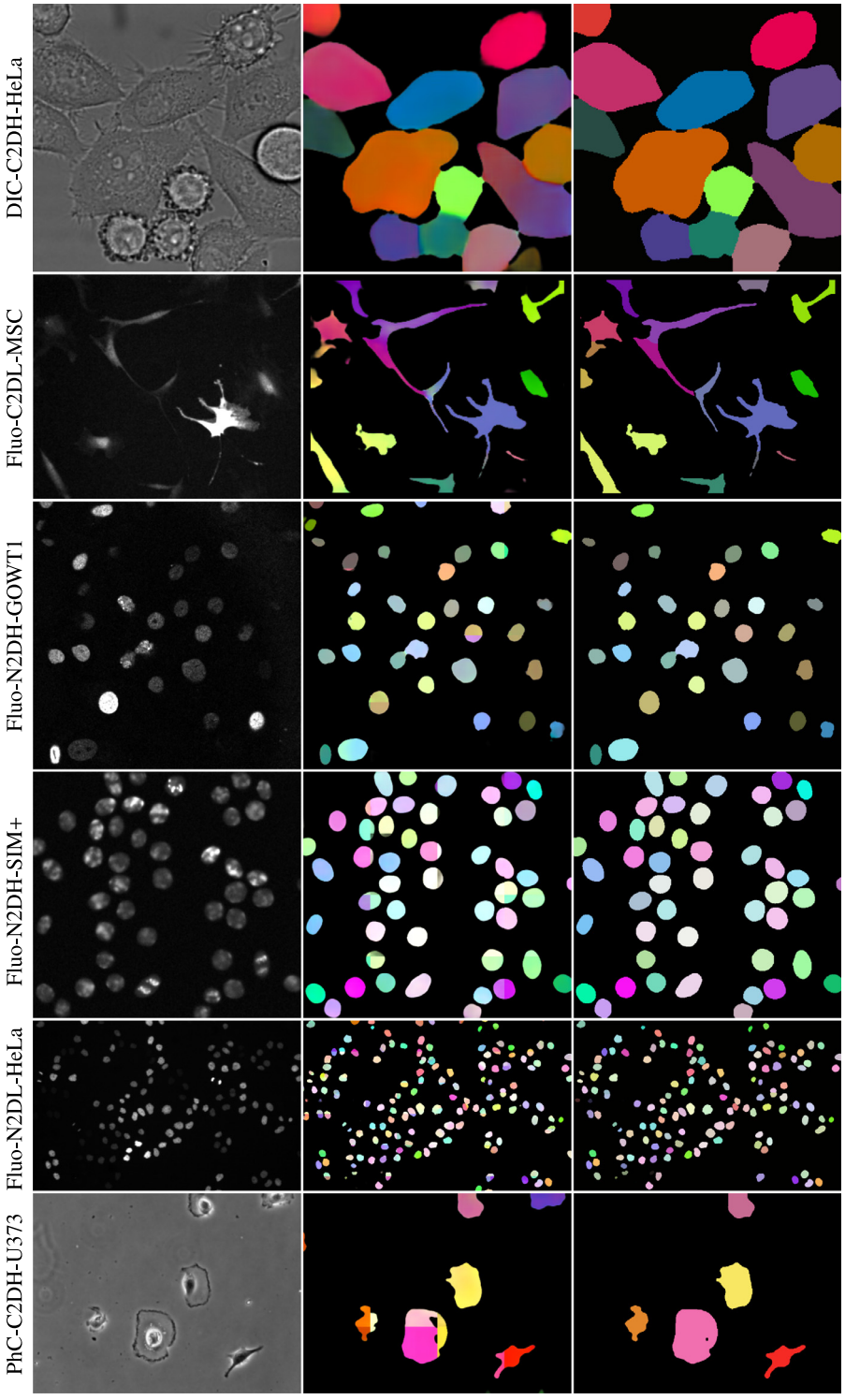


Fig. 8. Example results of the evaluated ISBI celltracking datasets. Left: input image; middle: three randomly chosen dimensions of the embedding vectors as RGB channels; right: final instance segmentation. We use the merging strategy for overlapping tiles as described in Section 2.4. However, for visualization purposes of the embedding vectors, we take the maximum response over all overlapping tiles, visible as straight edges inside instances.

Furthermore, in the tracking metric, our method ranks first in two datasets, and second in three datasets, confirming that our method well incorporates temporal information. For the dataset DIC-C2DH-HeLa, which has a dense layout of cells, we improve the results from our MICCAI work (Payer et al., 2018b) by 3% and outperform all other methods by a large margin in both tracking and segmentation metrics. On the dataset Fluo-N2DH-GOWT1 we rank overall second, improving our previously reported results by $\approx 1\%$. On the datasets Fluo-N2DL-HeLa and Fluo-N2DH-SIM+, the results for the overall performance metric improved by $\approx 10\%$ and $\approx 6\%$, respectively. Compared with our previous work, the improvements in the datasets Fluo-N2DH-GOWT1, Fluo-N2DL-HeLa, and Fluo-N2DH-SIM+ are mainly due to our proposed tiling strategy (see Section 2.4), where testing images are cropped into

Table 3
Results of the fold cross validation experiments comparing the proposed recurrent stacked hourglass network (RSHN) for individual mean shift bandwidth parameters h and the RSHN using HDBSCAN, as well as the stacked hourglass network using volumetric kernels (Volumetric). The values show the mean SEG, TRA and OP metric of both videos.

Network	Clustering Parameters	DIC-C2DH-HeLa			Fluo-N2DH-GOWT1			PhC-C2DH-U373			Overall Mean		
		SEG	TRA	OP	SEG	TRA	OP	SEG	TRA	OP	SEG	TRA	OP
RSHN	$h = 0.025$	**0.781**	**0.933**	**0.857**	0.777	0.975	0.876	0.722	0.935	0.828	0.760	0.948	0.854
	$h = 0.05$	**0.781**	0.932	**0.857**	0.776	**0.976**	0.876	0.729	0.942	0.835	0.762	0.950	0.856
	$h = 0.1$	0.780	0.924	0.852	0.777	**0.976**	**0.877**	0.748	**0.951**	0.849	0.768	**0.951**	**0.859**
	$h = 0.15$	0.773	0.906	0.840	**0.780**	**0.977**	**0.879**	**0.753**	**0.952**	**0.853**	**0.769**	0.945	0.857
	$h = 0.2$	0.685	0.845	0.765	0.778	0.976	0.877	**0.753**	0.951	0.852	0.739	0.924	0.831
	HDBSCAN	0.751	0.898	0.824	0.774	0.969	0.872	0.750	0.949	0.849	0.758	0.939	0.848
Volumetric	$h = 0.1$	0.735	0.918	0.827	0.779	0.973	0.876	0.626	0.929	0.778	0.713	0.940	0.827

Table 4
Quantitative results[5] of the ISBI celltracking challenge datasets for overall performance (OP), segmentation (SEG), and tracking (TRA), as described in Ulman et al. (2017). The superscripts (1)-(4) are used for different algorithms submitted by the same group.

		DIC-C2DH-HeLa	Fluo-C2DL MSC	Fluo-N2DH-GOWT1	Fluo-N2DL-HeLa	PhC-C2DH-U373	Fluo-N2DH-SIM+
OP	1st	0.894	0.759(1)	0.951(1)	0.944	0.951	0.884(4)
	2nd	0.845	0.676(1)	0.923	0.942(1)	0.948	0.882(2)
	3rd	0.834(4)	0.649	0.902	0.940	0.936	0.881
			0.625 5th		0.930 5th	0.900 5th	0.869 7th
SEG	1st	0.834	0.645(1)	0.927(1)	0.903	0.924	0.811(4)
	2nd	0.793(4)	0.617	0.894	0.902	0.922	0.807
	3rd	0.792	0.590(1)	0.893	0.900	0.920	0.802(2)
			0.462 7th	0.866 6th	0.879 5th	0.822 7th	0.765 8th
TRA	1st	0.954	0.873(1)	0.979	0.991(1)	0.981	0.975
	2nd	0.898	0.788	0.976(1)	0.988	0.978	0.973
	3rd	0.881	0.763(1)	0.932(2)	0.986	0.977(3)	0.966(3)
					0.982 5th		

Ours
BGU-IL(1)–(4)
CUNI-CZ
CVUT-CZ
FR-Fa-GE
FR-Ro-GE
HD-Har-GE
KIT-GE
KTH-SE(1)–(4)
LEID-NL

overlapping tiles and processed one after the other, thus removing the need to downsample the input images by a large factor.

Since in the ISBI celltracking challenge only a small number of images are fully segmented, while in the majority of the images cell instances are annotated with small dots in the center, our loss mainly focuses on assigning correct embeddings to the center of the cells. Thus, when compared to the center of the cells, the delineation of their borders, which is important for the segmentation metric, is learned from a lower amount of data. This might be the reason why only for one dataset our method is in the top three for the segmentation metric. Nevertheless, additionally to instance segmentation, our loss simultaneously optimizes for instance tracking in order to detect mitosis events, achieving overall the best results in the tracking metric.

6. Discussion and conclusion

In this paper we proposed a method that performs instance segmentation by representing instances as embedding vectors. Furthermore, it performs tracking of segmented instances by incorporating temporal information into a fully convolutional neural network architecture. Extending our preliminary work (Payer et al., 2018b), we improved our framework with a simpler clustering algorithm, as well as an image tiling strategy allowing arbitrary input image resolutions without the need for large downsampling factors. We show the wide applicability of our method on an in-house dataset involving ten high resolution H&E-stained microscopy images from sheep vocal cord muscle fibers, as well as on six diverse datasets of cell microscopy videos from the ISBI celltracking challenge (Ulman et al., 2017), where we show state-of-the-art results and improvements as compared to our previous work (Payer et al., 2018b).

Our work contributes with an embedding loss that directly predicts instances represented in an embedding space, which is based on cosine similarities. Our experiments show that this loss is well suitable for cell instance segmentation and tracking, e.g., in Section 5.1 we show that the loss function performs better than a weighted softmax cross entropy loss, especially for dense cells. Furthermore, benefiting from the bounded and normalized values used for calculating cosine similarities, we successfully combined the cosine embedding loss with recurrent neural networks to perform tracking of embeddings. In contrast to the recent work of Kong and Fowlkes (2018), who use an embedding loss based on cosine similarities for segmenting a small number of instances from a static image, our loss function can also be used for tracking and does not require all instances to be dissimilar, but only neighboring ones. As the number of representable instances is limited by the dimension of the embedding vectors, the requirement of only neighboring instances being dissimilar is necessary when predicting a possibly large number of instances, as e.g., is the case in cell microscopy images.

Our work also contributes to cell instance segmentation and tracking with the network architecture RSHN, which integrates temporal information with ConvGRUs. We show that integrating temporal information as states implemented with ConvGRUs produces better results when compared to integrating temporal information as an additional spatial dimension for volumetric networks (see Section 5.2). Furthermore, compared to our preliminary work (Payer et al., 2018b), we simplify the clustering of embedding vectors by exchanging HDBSCAN with mean shift, which produces better results, despite solely needing a single parameter that

is set equal for each dataset (see Section 5.3). Additionally, our proposed tiling strategy allows processing images with high resolution, which is shown, e.g., on the large resolution H&E-stained microscopy images (see Section 5.1). Due to all these contributions, we improve upon our previous work (Payer et al., 2018b), and achieve state-of-the-art results on six datasets from the ISBI celltracking challenge (see Section 5.4). However, it has to be noted that the cell movements in the ISBI celltracking challenge are relatively small with a large overlapping area of the cell instances in consecutive time points. Therefore, for datasets with larger movements of instances, further modification of our method might be needed.

In conclusion, we have shown that predicting embedding vectors for instance segmentation can be successfully combined with incorporating temporal information as recurrent networks for instance tracking. In future work, we will investigate how to effectively incorporate mean shift into our network, to allow end-to-end training. Furthermore, we plan to extend our method to work with high resolution volumetric images of dense instances, e.g., Berning et al. (2015), as well as volumetric video sequences, e.g., Ulman et al. (2017).

Acknowledgments

We thank Claus Gerstenberger and Markus Gugatschka for providing the muscle fiber dataset and supervising its annotation, as well as Saban Öztürk and Thomas Neff for helping with our experiments. We gratefully acknowledge the support of NVIDIA Corporation with the donation of the Titan V GPU used in this research. This work was supported by the Austrian Research Promotion Agency, funds no. 848458 and no. 871262. Furthermore, this work was supported by the Austrian Science Fund (FWF): P28078-N33.

References

Akram, S.U., Kannala, J., Eklund, L., Heikkilä, J., 2016. Cell segmentation proposal network for microscopy image analysis. In: Deep Learn. Data Labeling Med. Appl.. Springer International Publishing, Cham, pp. 21–29.

Appel, K.I., Haken, W., 1976. Every planar map is four colorable. Bull. Am. Math. Soc. 82 (5), 711–712.

Arbelle, A., Reyes, J., Chen, J.-Y., Lahav, G., Raviv, T.R., 2018. A probabilistic approach to joint cell tracking and segmentation in high-throughput microscopy videos. Med. Image Anal. 47, 140–152. doi:10.1016/j.media.2018.04.006.

Ballas, N., Yao, L., Pal, C., Courville, A., 2016. Delving deeper into convolutional networks for learning video representations. Int. Conf. Learn. Represent.. CoRR, abs:1511.06432

Bennie, S., Petrofsky, J., Nisperos, J., Tsurudome, M., Laymon, M., 2002. Toward the optimal waveform for electrical stimulation of human muscle. Eur. J. Appl. Physiol. 88 (1–2), 13–19. doi:10.1007/s00421-002-0711-4.

Bensch, R., Ronneberger, O., 2015. Cell segmentation and tracking in phase contrast images using graph cut with asymmetric boundary costs. In: IEEE Int. Symp. Biomed. Imaging. IEEE, pp. 1220–1223. doi:10.1109/ISBI.2015.7164093.

Bergeest, J.-P., Rohr, K., 2012. Efficient globally optimal segmentation of cells in fluorescence microscopy images using level sets and convex energy functionals. Med. Image Anal. 16 (7), 1436–1444. doi:10.1016/j.media.2012.05.012.

Berning, M., Boergens, K.M., Helmstaedter, M., 2015. Segem: efficient image analysis for high-Resolution connectomics. Neuron 87 (6), 1193–1206. doi:10.1016/j.neuron.2015.09.003.

Campello, R.J.G.B., Moulavi, D., Zimek, A., Sander, J., 2015. Hierarchical density estimates for data clustering, visualization, and outlier detection. ACM Trans. Knowl. Discov. Data 10 (1), 5:1–5:51. doi:10.1145/2733381.

Carpenter, A.E., Jones, T.R., Lamprecht, M.R., Clarke, C., Kang, I.H., Friman, O., Guertin, D.A., Chang, J.H., Lindquist, R.A., Moffat, J., Golland, P., Sabatini, D.M., 2006. Cellprofiler: image analysis software for identifying and quantifying cell phenotypes. Genome Biol. 7 (10), R100. doi:10.1186/gb-2006-7-10-r100.

Chen, H., Qi, X., Yu, L., Dou, Q., Qin, J., Heng, P.-A., 2017. DCAN: Deep contour-aware networks for object instance segmentation from histology images. Med. Image Anal. 36, 135–146. doi:10.1016/j.media.2016.11.004.

Cho, K., Van Merriënboer, B., Gulcehre, C., Bahdanau, D., Bougares, F., Schwenk, H., Bengio, Y., 2014. Learning phrase representations using RNN encoder-decoder for statistical machine translation. Proc. Empir. Methods Nat. Lang. Process. 1724–1734.

Ciresan, D.C., Giusti, A., Gambardella, L.M., Schmidhuber, J., 2013. Mitosis detection in breast cancer histology images with deep neural networks. In: Proc. Med. Image Comput. Comput. Interv.. Springer Berlin Heidelberg, Berlin, Heidelberg, pp. 411–418. doi:10.1007/978-3-642-40763-5_51.

Comaniciu, D., Meer, P., 2002. Mean shift: a robust approach toward feature space analysis. IEEE Trans. Pattern Anal. Mach. Intell. 24 (5), 603–619. doi:10.1109/34.1000236.

Condeelis, J., Pollard, J.W., 2006. Macrophages: obligate partners for tumor cell migration, invasion, and metastasis. Cell 124 (2), 263–266. doi:10.1016/j.cell.2006.01.007.

Evans, R., Patzak, I., Svensson, L., De Filippo, K., Jones, K., McDowall, A., Hogg, N., 2009. Integrins in immunity. J. Cell Sci. 122 (2), 215–225. doi:10.1242/jcs.019117.

Girshick, R., Donahue, J., Darrell, T., Malik, J., 2014. Rich feature hierarchies for accurate object detection and semantic segmentation. In: Proc. Comput. Vis. Pattern Recognit., pp. 580–587. doi:10.1109/CVPR.2014.81.

Glorot, X., Bordes, A., Bengio, Y., 2011. Deep sparse rectifier neural networks. In: Proc. Int. Conf. Artif. Intell. Stat., pp. 315–323.

Graham, S., Chen, H., Gamper, J., Dou, Q., Heng, P.-A., Snead, D., Tsang, Y.W., Rajpoot, N.M., 2019. MILD-Net: Minimal information loss dilated network for gland instance segmentation in colon histology images. Med. Image Anal. 52, 199–211. doi:10.1016/j.media.2018.12.001.

Gurcan, M., Boucheron, L., Can, A., Madabhushi, A., Rajpoot, N.M., Yener, B., 2009. Histopathological image analysis: A Review. IEEE Rev. Biomed. Eng. 2, 147–171. doi:10.1109/RBME.2009.2034865.

Harley, A.W., Derpanis, K.G., Kokkinos, I., 2017. Segmentation-aware convolutional networks using local attention masks. In: Proc. Int. Conf. Comput. Vis., pp. 5038–5047.

He, K., Gkioxari, G., Dollár, P., Girshick, R., 2017. Mask R-CNN. In: Proc. Int. Conf. Comput. Vis., pp. 2980–2988.

He, K., Zhang, X., Ren, S., Sun, J., 2015. Delving deep into rectifiers: surpassing human-Level performance on ImageNet classification. In: Proc. Int. Conf. Comput. Vis.. IEEE, pp. 1026–1034. doi:10.1109/ICCV.2015.123.

Hochreiter, S., Schmidhuber, J., 1997. Long short-Term memory. Neural Comput. 9 (8), 1735–1780. doi:10.1162/neco.1997.9.8.1735.

Irshad, H., Veillard, A., Roux, L., Racoceanu, D., 2014. Methods for nuclei detection, segmentation, and classification in digital histopathology: A Review - Current status and future potential. IEEE Rev. Biomed. Eng. 7, 97–114. doi:10.1109/RBME.2013.2295804.

Kainz, P., Pfeiffer, M., Urschler, M., 2017. Segmentation and classification of colon glands with deep convolutional neural networks and total variation regularization. PeerJ 5, e3874. doi:10.7717/peerj.3874.

Karbiener, M., Jarvis, J.C., Perkins, J.D., Lanmüller, H., Schmoll, M., Rode, H.S., Gerstenberger, C., Gugatschka, M., 2016. Reversing age related changes of the laryngeal muscles by chronic electrostimulation of the recurrent laryngeal nerve. PLoS ONE 11 (11), e0167367. doi:10.1371/journal.pone.0167367.

Kingma, D.P., Ba, J., 2015. Adam: A Method for stochastic optimization. Int. Conf. Learn. Represent. doi:10.1145/1830483.1830503. CoRR, abs:1412.6980

Kong, S., Fowlkes, C., 2018. Recurrent Pixel Embedding for Instance Grouping. In: Proc. Comput. Vis. Pattern Recognit., pp. 9018–9028.

Kraus, O.Z., Ba, J.L., Frey, B.J., 2016. Classifying and segmenting microscopy images with deep multiple instance learning. Bioinformatics 32 (12), i52–i59. doi:10.1093/bioinformatics/btw252.

LeCun, Y., Bengio, Y., Hinton, G., 2015. Deep learning. Nature 521 (7553), 436–444. doi:10.1038/nature14539.

LeCun, Y., Bottou, L., Bengio, Y., Haffner, P., 1998. Gradient-based learning applied to document recognition. Proc. IEEE 86 (11), 2278–2323. doi:10.1109/5.726791.

Li, C., Wang, X., Liu, W., Latecki, L.J., 2018. Deepmitosis: mitosis detection via deep detection, verification and segmentation networks. Med. Image Anal. 45, 121–133. doi:10.1016/j.media.2017.12.002.

Magnusson, K.E.G., Jalden, J., 2012. A batch algorithm using iterative application of the Viterbi algorithm to track cells and construct cell lineages. In: IEEE Int. Symp. Biomed. Imaging. IEEE, pp. 382–385. doi:10.1109/ISBI.2012.6235564.

Maška, M., Ulman, V., Svoboda, D., Matula, P., Matula, P., Ederra, C., Urbiola, A., España, T., Venkatesan, S., Balak, D.M., Karas, P., Bolcková, T., Štreitová, M., Carthel, C., Coraluppi, S., Harder, N., Rohr, K., Magnusson, K.E., Jaldén, J., Blau, H.M., Dzyubachyk, O., KÁížek, P., Hagen, G.M., Pastor-Escuredo, D., Jimenez-Carretero, D., Ledesma-Carbayo, M.J., Muñoz-Barrutia, A., Meijering, E., Kozubek, M., Ortiz-De-Solorzano, C., 2014. A benchmark for comparison of cell tracking algorithms. Bioinformatics 30 (11), 1609–1617. doi:10.1093/bioinformatics/btu080.

Meijering, E., 2012. Cell segmentation: 50 years down the road. IEEE Signal Process. Mag. 29 (5), 140–145. doi:10.1109/MSP.2012.2204190.

Montell, D.J., 2008. Morphogenetic cell movements: diversity from modular mechanical properties. Science 322 (5907), 1502–1505. doi:10.1126/science.1164073.

Newell, A., Huang, Z., Deng, J., 2017. Associative embedding: end-to-end learning for joint detection and grouping. In: Adv. Neural Inf. Process. Syst.. Curran Associates, Inc., pp. 2277–2287.

Newell, A., Yang, K., Deng, J., 2016. Stacked hourglass networks for human pose estimation. In: Proc. Eur. Conf. Comput. Vis., pp. 483–499. doi:10.1007/978-3-319-46484-8.

Payer, C., Štern, D., Bischof, H., Urschler, M., 2018. Multi-label whole heart segmentation using CNNs and anatomical label configurations. In: Stat. Atlases Comput. Model. Hear. ACDC MMWHS Challenges.. Springer International Publishing, pp. 190–198. doi:10.1007/978-3-319-75541-0_20.

Payer, C., Štern, D., Bischof, H., Urschler, M., 2019. Integrating spatial configuration into heatmap regression based CNNs for landmark localization. Med. Image Anal. 54, 207–219. doi:10.1016/j.media.2019.03.007.

Payer, C., Štern, D., Neff, T., Bischof, H., Urschler, M., 2018. Instance segmentation and tracking with cosine embeddings and recurrent hourglass networks. In: Proc. Med. Image Comput. Comput. Interv.. Springer International Publishing, Cham, pp. 3–11. doi:10.1007/978-3-030-00934-2_1.

Raza, S.E.A., Cheung, L., Shaban, M., Graham, S., Epstein, D., Pelengaris, S., Khan, M., Rajpoot, N.M., 2019. Micro-Net: a unified model for segmentation of various objects in microscopy images. Med. Image Anal. 52, 160–173. doi:10.1016/j.media.2018.12.003.

Rempfler, M., Stierle, V., Ditzel, K., Kumar, S., Paulitschke, P., Andres, B., Menze, B.H., 2018. Tracing cell lineages in videos of lens-free microscopy. Med. Image Anal. 48, 147–161. doi:10.1016/j.media.2018.05.009.

Ren, M., Zemel, R.S., 2017. End-To-End instance segmentation with recurrent attention. In: Proc. Comput. Vis. Pattern Recognit., pp. 6656–6664.

Romera-Paredes, B., Torr, P.H.S., 2016. Recurrent instance segmentation. In: Proc. Eur. Conf. Comput. Vis., pp. 312–329. doi:10.1007/978-3-319-46466-4_19.

Ronneberger, O., Fischer, P., Brox, T., 2015. U-Net: Convolutional networks for biomedical image segmentation. In: Proc. Med. Image Comput. Comput. Interv.. Springer, pp. 234–241. doi:10.1007/978-3-319-24574-4_28.

Schiegg, M., Hanslovsky, P., Haubold, C., Koethe, U., Hufnagel, L., Hamprecht, F.A., 2015. Graphical model for joint segmentation and tracking of multiple dividing cells. Bioinformatics 31 (6), 948–956. doi:10.1093/bioinformatics/btu764.

Shelhamer, E., Long, J., Darrell, T., 2017. Fully convolutional networks for semantic segmentation. IEEE Trans. Pattern Anal. Mach. Intell. 39 (4), 640–651. doi:10.1109/TPAMI.2016.2572683.

Sirinukunwattana, K., Pluim, J.P.W., Chen, H., Qi, X., Heng, P.A., Guo, Y.B., Wang, L.Y., Matuszewski, B.J., Bruni, E., Sanchez, U., Böhm, A., Ronneberger, O., Cheikh, B.B., Racoceanu, D., Kainz, P., Pfeiffer, M., Urschler, M., Snead, D.R.J., Rajpoot, N.M., 2017. Gland segmentation in colon histology images: the glas challenge contest. Med. Image Anal. 35, 489–502. doi:10.1016/j.media.2016.08.008.

Song, Y., Tan, E.-L., Jiang, X., Cheng, J.-Z., Ni, D., Chen, S., Lei, B., Wang, T., 2017. Accurate cervical cell segmentation from overlapping clumps in pap smear images. IEEE Trans. Med. Imaging 36 (1), 288–300. doi:10.1109/TMI.2016.2606380.

Tao, X., Gao, H., Liao, R., Wang, J., Jia, J., 2017. Detail-revealing deep video super-resolution. In: Proc. Int. Conf. Comput. Vis., pp. 4472–4480.

Tokmakov, P., Alahari, K., Schmid, C., 2017. Learning video object segmentation with visual memory. In: Proc. Int. Conf. Comput. Vis., pp. 4481–4490. doi:10.1109/ICCV.2017.480.

Türetken, E., Wang, X., Becker, C.J., Haubold, C., Fua, P., 2017. Network flow integer programming to track elliptical cells in time-Lapse sequences. IEEE Trans. Med. Imaging 36 (4), 942–951. doi:10.1109/TMI.2016.2640859.

Ulman, V., Maška, M., Magnusson, K.E., Ronneberger, O., Haubold, C., Harder, N., Matula, P., Matula, P., Svoboda, D., Radojevic, M., Smal, I., Rohr, K., Jaldén, J., Blau, H.M., Dzyubachyk, O., Lelieveldt, B., Xiao, P., Li, Y., Cho, S.Y., Dufour, A.C., Olivo-Marin, J.C., Reyes-Aldasoro, C.C., Solis-Lemus, J.A., Bensch, R., Brox, T., Stegmaier, J., Mikut, R., Wolf, S., Hamprecht, F.A., Esteves, T., Quelhas, P., Demirel, Ö., Malmström, L., Jug, F., Tomancak, P., Meijering, E., Muñoz-Barrutia, A., Kozubek, M., Ortiz-De-Solorzano, C., 2017. An objective comparison of cell-tracking algorithms. Nat. Methods 14 (12), 1141–1152. doi:10.1038/nmeth.4473.

Veta, M., van Diest, P.J., Kornegoor, R., Huisman, A., Viergever, M.A., Pluim, J.P.W., 2013. Automatic nuclei segmentation in h&e stained breast cancer histopathology images. PLoS ONE 8 (7), e70221. doi:10.1371/journal.pone.0070221.

Wählby, C., Sintorn, I.M., Erlandsson, F., Borgefors, G., Bengtsson, E., 2004. Combining intensity, edge and shape information for 2D and 3D segmentation of cell nuclei in tissue sections. J. Microsc. 215 (1), 67–76. doi:10.1111/j.0022-2720.2004.01338.x.

Xie, Y., Xing, F., Shi, X., Kong, X., Su, H., Yang, L., 2018. Efficient and robust cell detection: a structured regression approach. Med. Image Anal. 44, 245–254. doi:10.1016/j.media.2017.07.003.

Xingjian, S., Chen, Z., Wang, H., Yeung, D.-Y., Wong, W.-K., Woo, W.-c., 2015. Convolutional LSTM network: A machine learning approach for precipitation nowcasting. In: Adv. Neural Inf. Process. Syst., pp. 802–810.

Xu, Y., Li, Y., Wang, Y., Liu, M., Fan, Y., Lai, M., Chang, E.I., 2017. Gland instance segmentation using deep multichannel neural networks. IEEE Trans. Biomed. Eng. 64 (12), 2901–2912. doi:10.1109/TBME.2017.2686418.

Zimmer, C., Zhang, B., Dufour, A., Thebaud, A., Berlemont, S., Meas-Yedid, V., Marin, J.-C., 2006. On the digital trail of mobile cells. IEEE Signal Process. Mag. 23 (3), 54–62. doi:10.1109/MSP.2006.1628878.