VoxelEmbed: 3D Instance Segmentation and Tracking with Voxel Embedding based Deep Learning

Mengyang Zhao[1], Quan Liu[2], Aadarsh Jha[2], Ruining Deng[2], Tianyuan Yao[2], Anita Mahadevan-Jansen[2], Matthew J. Tyska[2], Bryan A. Millis[2], and Yuankai Huo[2(✉)]

[1] Dartmouth College, Hanover, NH 03755, USA
[2] Vanderbilt University, Nashville, TN 37215, USA
yuankai.huo@vanderbilt.edu

Abstract. Recent advances in bioimaging have provided scientists a superior high spatial-temporal resolution to observe dynamics of living cells as 3D volumetric videos. Unfortunately, the 3D biomedical video analysis is lagging, impeded by resource insensitive human curation using off-the-shelf 3D analytic tools. Herein, biologists often need to discard a considerable amount of rich 3D spatial information by compromising on 2D analysis via maximum intensity projection. Recently, pixel embedding based cell instance segmentation and tracking provided a neat and generalizable computing paradigm for understanding cellular dynamics. In this work, we propose a novel spatial-temporal voxel-embedding (VoxelEmbed) based learning method to perform simultaneous cell instance segmenting and tracking on 3D volumetric video sequences. Our contribution is in four-fold: (1) The proposed voxel embedding generalizes the pixel embedding with 3D context information; (2) Present a simple multi-stream learning approach that allows effective spatial-temporal embedding; (3) Accomplished an end-to-end framework for one-stage 3D cell instance segmentation and tracking without heavy parameter tuning; (4) The proposed 3D quantification is memory efficient via a single GPU with 12 GB memory. We evaluate our VoxelEmbed method on four 3D datasets (with different cell types) from the ISBI Cell Tracking Challenge. The proposed VoxelEmbed method achieved consistent superior overall performance (OP) on two densely annotated datasets. The performance is also competitive on two sparsely annotated cohorts with 20.6% and 2% of data-set having segmentation annotations. The results demonstrate that the VoxelEmbed method is a generalizable and memory-efficient solution.

Keywords: Tracking · Segmentation · Instances · 3-D · Embedding

1 Introduction

Characterizing cellular dynamics, the behaviors of the fundamental units of life, is indispensable in translational biological research, such as organogenesis [4], immune response [24], drug development [31], and cancer metastasis [7]. With a tenet of "seeing is believing", recent advances in bioimaging have provided scientists unprecedented high spatial-temporal resolution to observe three dimensional dynamics (3D volumes

C. Lian et al. (Eds.): MLMI 2021, LNCS 12966, pp. 437–446, 2021.
https://doi.org/10.1007/978-3-030-87589-3_45

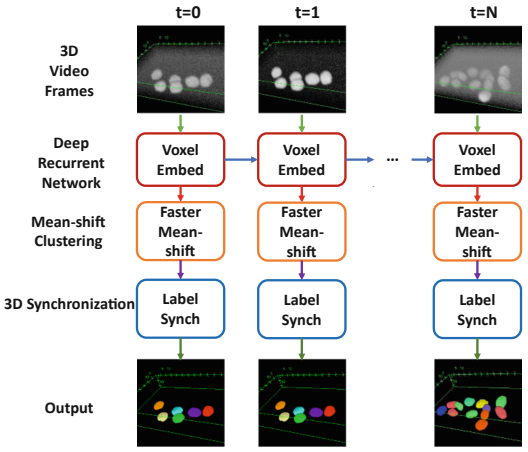

Fig. 1. The overall framework. The workflow of the proposed voxel embedding based deep learning framework is presented, for 3D cell instance segmentation and tracking.

+ time) of living cells [16]. Yet, large-scale bioimage data are concurrent with imaging innovations, causing fundamental computational challenges for quantifying cellular dynamics in translational biological research, where "quantifying is deciding" [19]. For example, a single lattice light-sheet microscope [6] (LLS) produce TB level rich spatial-temporal dynamic 3D volumetric videos [27]. Unfortunately, the 3D biomedical video analysis is lagging, impeded by resource insensitive human curation via off-the-shelf 3D analytic tools [5]. Recent deep learning techniques have achieved remarkable success in computer vision and biomedical image analysis. However, the large-scale quantification of "3D+time" cellular dynamics with deep learning (e.g., dense instance segmentation and tracking) is still hindered by the high dimensionality and heterogeneity of the dynamics. [11] Herein, biologists often need to discard a considerable amount of rich 3D spatial information by projecting the 3D videos to 2D space for downstream analyses.

When quantifying cellular dynamics, the "segment-then-track" two-stage paradigm [7,8,13,28,29] is a prevalent design for both conventional model based methods and deep learning approaches. Such a paradigm first segments instance objects across frames and then links the instance objects via association algorithms. However, the spatial and temporal information from the same individual object cannot be learned simultaneously with the two-stage design. To integrate and handle these two tasks simultaneously, Payer et al. [25] proposed a cosine pixel embedding based recurrent stacked hourglass network (RSHN) for simultaneous instance cell segmentation and tracking. The pixel embedding approach tackled instance segmentation and tracking within an uniformed "single-stage" framework. The key idea is to loosen the constraint of having each cell requiring a globally unique embedding, to just allowing the cell to have a different embedding relative to the nearby four cells (based on the Four Color Map theorem [1,25]). Herein, the number of embeddings does not have to strictly increase with the number of cells, providing a scalable learning strategy. Based on the merits, it is appealing to adapt such strategy from 2D to 3D settings. However, the direct 3D

adaptation (e.g., change a 2D network to a 3D version) requires higher computational resources, more training samples, and a considerably larger embedding space to differentiate each cell to all of its neighbors in 3D.

In this study, we propose a novel spatial-temporal voxel-embedding (VoxelEmbed) deep learning based method to perform 3D cell instance segmentation and tracking. As opposed to deploying a 3D network, we develop a simple multi-stream learning approach to learn spatial, temporal, and 3D context information simultaneously via a 2D network design. To aggregate 3D information, we introduce a 3D synchronization algorithm to build volumetric masks inspired by recent advances in slice propagation [3]. Briefly, the innovations of the proposed approach is in four-folds:

(1) The proposed VoxelEmbed approach generalizes the pixel embedding to a voxel embedding with 3D context information.
(2) A simple multi-stream learning approach is presented that allows effective spatial-temporal embedding.
(3) To our knowledge, this is the first embedding based deep learning approach for simultaneous 3D cell instance segmentation and tracking.
(4) The proposed method is memory efficient to a single GPU with 12 GB memory.

2 Methods

The principle of our proposed VoxelEmbed framework is presented Fig. 1. In VoxelEmbed, we extend spatial-temporal embedding by adding 3D information. The extra embedding information is obtained from the zigzag multi-stream straining.

2.1 Cosine Embedding Based Instance Segmentation and Tracking

Payer et al. [25] proposed a cosine embedding based recurrent stacked hourglass network (RSHN) that, for the first time, integrated the instance segmentation and tracking algorithms into a one-stage holistic learning framework with pixel-wise cosine embedding, which achieves instance cell segmentation and tracking in a one-stage framework. The entire pixel embedding based learning consist of two major stages: embedding encoding and feature clustering.

The RSHN network, combining convolutional GRUs [2,12] (ConvGRUs) and the stacked hourglass network [21], is employed as the backbone network for our voxel embedding. In [25], each pixel in a 2D video sequence was encoded to a high-dimensional embedding vector with the intuition that all pixels from the same cell, across spatial and temporal, should have the same feature representation (embedding). The renown cosine similarity [14,15] approach is commonly used to measure the similarities of any two pixels. For instance, $\mathbf{A}$ was the embedding vector of pixel a, and $\mathbf{B}$ was the embedding vector of pixel b. The cosine similarity of $\mathbf{A}$ and $\mathbf{B}$ is defined as:

$$cos\,(\mathbf{A},\mathbf{B}) = \frac{\mathbf{A}\cdot\mathbf{B}}{\|\mathbf{A}\|\ \|\mathbf{B}\|} \tag{1}$$

which ranged from -1 to 1, where 1 indicates that two vectors have the same direction, 0 indicates orthogonal, and -1 indicates the opposite. The cosine similarities is used as a loss function to force the pixels from the same cell to have $cos\,(\mathbf{A},\mathbf{B})$ towards 1, while the pixels from different cells to be 0.

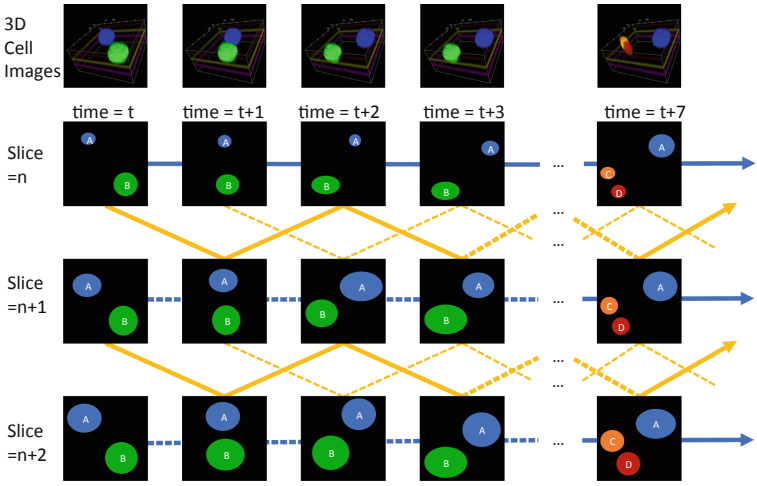

Fig. 2. VoxelEmbed at the training stage. Our VoxelEmbed model is trained by both blue and yellow streams, to ensure the same voxel embedding of the same object in terms of spatial-temporal domain and 3D context.

2.2 Voxel Embedding (Training Stage)

To encode the 3D context information for 3D cell instance segmentation and tracking, we generalize the pixel embedding principle to a voxel embedding scheme. Briefly, the VoxelEmbed approach learns additional 3D context features, and concatenates those features to original pixel embedding features. The additional 3D context features are learned from the new zigzag training path (the golden stream in Fig. 2) beyond the temporal embedding path (the blue stream in Fig. 2).

The sections with the same z-axis location from all 3D volumes across the temporal direction are used as the temporal embedding path, to ensure the same embedding of each cell (at the same z-axis location in 2D) has the same embedding along with migration and time.

Then, the sections at nearby z-axis locations are selected in a zigzag training path to enforce the embedding similarity along with the migration and 3D context, when learning all zigzag training paths converge the entire 3D space. The zigzag training paths are designed as an interleaved "W" shape. Using multi-stream training, the voxels from the same cell in a 3D video will be forced to have the same embedding along with migration, time, and 3D context.

2.3 Voxel Embedding (Testing Stage)

In implementation, T frames from time t to $t+T$ are used as a single training sample. The temporal embedding feature, a 14-dimensional feature vector, is encoded for each voxel:

$$f_{time} = [a_1, a_2, ..., a_{14}] \tag{2}$$

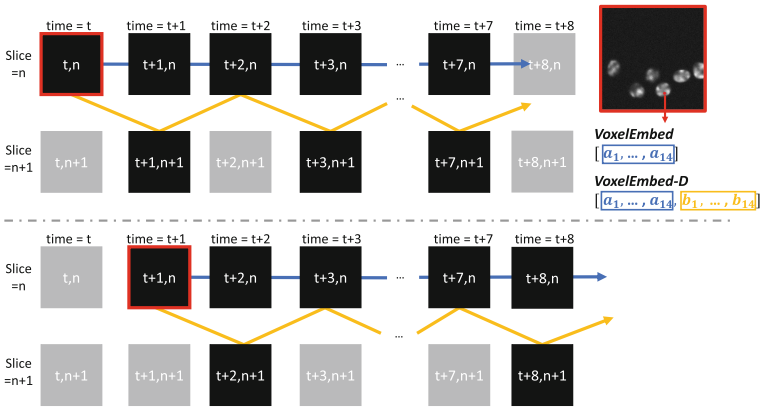

Fig. 3. VoxelEmbed at the testing stage. The blue and yellow streams cover two ways of forming testing samples (e.g., T = 8 in this study) for obtaining embedding of the first frame (red boarder) at the testing stage. With the embedded feature maps, the 3D cell instance segmentation and tracking results are obtained from the 3D synchronization (see Algorithm 1).

With the same principle, T frames from time t to $t + T$ with zigzag frames are used as another single training example. The 3D context embedding feature, another 14-dimensional feature vector, is encoded for each voxel:

$$f_{threeD} = [b_1, b_2, ..., b_{14}] \tag{3}$$

Then, the f_{time} and f_{threeD} are concatenated into a 28 dimension feature for each voxel as our final voxel embedding feature:

$$f_{spatial-temporal} = f_{time} \oplus f_{threeD} = [a_1, ..., a_{14}, b_1, ..., b_{14}] \tag{4}$$

where $\oplus$ is the concatenation operator.

In the testing stage, the encoded embedding images will be clustered to each individual cells. The standard Mean-shift clustering algorithm was used in Payer et al. [25]. In this study, we utilize a GPU accelerated Faster Mean-shift algorithm [30] to accelerate the embedding clustering. From the unsupervised clustering, a unique label will be assigned to each cell as different instances, which achieve the instance segmentation and tracking simultaneously.

2.4 3D Synchronization

In the original pixel-embedding approach, the temporal synchronization step was introduced to stick different short "video clips" to the a complete full cell video, with consistent instance numbers. Inspired by this approach, we extend the synchronization from temporal to spatial as well. To do so, we propose a 3-D synchronization Algorithm 1 to improve label synchronization on the z-axis direction.

Algorithm 1. 3D Synchronization Algorithm

Require: $InMasks$: Unsynchronized mask-set; $ReMasks$: Reference mask-set;
Ensure: $OutMasks$: Synchronized mask-set;

for $mask \in InMasks$ **do**
  for $rmask \in ReMasks$ **do**
    $S_{mask} \leftarrow J(rmask, mask)$ # Caluate the Jaccard similarity: $J(R,S) = \frac{|R \cap S|}{|R \cup S|}$
    if S_{mask} is largest in the layer of $mask$ **then**
      $List_{rmask}$ append $mask$ # Get the largest similarity instance in each layers
    end if
  end for
end for
for $list \in List_{rmask}$ **do**
  Find the most common label $l \in list$
  for $mask \in list$ **do**
    $mask \leftarrow l$ # Synchronize the mask
  end for
end for

In Algorithm 1, the input reference mask-set $ReMasks$ is the layer with the largest foreground ratio. In Jaccard Similarity [22], R is the set of pixels belonging to reference mask and S is the set of pixels belonging to matching mask.

3 Data and Implementation Details

In this study, we conduct an empirical validation via the ISBI Cell Tracking Challenge [26] dataset to evaluate the accuracy performance of our proposed framework. Specifically, we used 3-D microscope video sequences from the ISBI Cell Tracking Challenge, which is independent with training data. The following four video sequence datasets of different sizes, shapes, and textures cells were adopted to evaluate the performance: (1) Chinese Hamster Ovarian nuclei(Fluo-N3DH-CHO), (2) C.elegans developing embryo (Fluo-N3DH-CE), (3) Simulated nuclei of HL60 cells(Fluo-N3DH-SIM+), and (4) Simulated GFP-actin-stained A549 Lung Cancer cells (Fluo-C3DH-A549-SIM). Note that the labels of the official testing data have not been released. Since each cohort has two training videos, we used one as training while another as testing. The source code of benchmarks (LEID-NL [9], KTH-SE [17], and RSHN [25]) was also deployed on such data directly. Therefore, the reported results could be different from the online leader board [10].

All computation and training were performed via a standard NC6 [20] virtual machine platform at the Microsoft Azure cloud. The virtual machine includes half an NVIDIA Tesla K80 accelerator [23] card (12 GB accessible) and six Intel Xeon E5-2690 v3 (Haswell) processor. The multi-stream training and 3-D Synchronization Algorithm was implemented with tensorflow and Python3. All the models in this study were trained with 20,000 iterations. During model training, the learning rate was initially set to 0.0001, and decreases to 0.00001 after 10,000 iterations. The bandwidth hyper-parameter in mean-shift clustering is set to 0.1.

Table 1. Quantitative result of empirical validation

Method	Data-set					
	Fluo-N3DH-SIM+			Fluo-C3DH-A549-SIM		
	SEG	TRA	OP	SEG	TRA	OP
LEID-NL [9]	0.643$^{(3)}$	**0.971**$^{(1)}$	0.807$^{(3)}$	0.827$^{(2)}$	1.000	0.914$^{(2)}$
KTH-SE [17]	0.774$^{(1)}$	0.958$^{(2)}$	0.866$^{(1)}$	0.842$^{(1)}$	1.000	0.921$^{(1)}$
RSHN [25]	0.748	0.909	0.829	0.798	1.000	0.899
VoxelEmbed (ours)	**0.818**	0.933	0.876	**0.852**	1.000	**0.926**
VoxelEmbed-D (ours)	0.806	0.952	**0.879**	0.838	1.000	0.919

* "(1), (2), (3)" indicates the current ranking in the Cell Tracking Challenge leader board [10].

Table 2. Quantitative result of empirical validation

Method	Data-set					
	Fluo-N3DH-CHO (20.6% frames have labels)			Fluo-N3DH-CE (2% frames have labels)		
	SEG	TRA	OP	SEG	TRA	OP
LEID-NL [9]	0.901$^{(4)}$	0.923$^{(5)}$	0.912$^{(5)}$	–	–	–
KTH-SE [17]	**0.907**$^{(2)}$	0.953$^{(1)}$	**0.930**$^{(1)}$	0.667$^{(3)}$	**0.945**$^{(1)}$	0.806$^{(2)}$
RSHN [25]	0.837	0.946	0.892	0.673	0.875	0.774
VoxelEmbed (ours)	0.862	0.958	0.910	**0.717**	0.897	0.807
VoxelEmbed-D (ours)	0.860	**0.959**	0.910	0.709	0.925	**0.817**

* "(1), (2), (3)" indicates the current ranking of such approach in the Cell Tracking Challenge leader board [10]. LEID did not provide N3DH-CE, the performance is not available (–).

4 Results

The qualitative results are shown in Fig. 4, while the comparison of quantitative results are presented in Table 1 and 2. In Table 1 and 2, the Cell Tracking Challenge's official tool of measuring the performance of tracking (TRA) and segmentation (SEG) are employed. TRA is computed by the normalized Acyclic Oriented Graph Matching measure (AOGM) [18] and is used as the tracking accuracy metric. The SEG is computed by Jaccard index. Following the ISBI Cell Tracking Challenge's Benchmark, we also compute the overall performance (OP), which is the average of TRA and SEG.

Based on quantitative and qualitative results, our methods achieved a competitive accuracy performance, using the same network without heavy parameter tuning. For Fluo-N3DH-SIM+ and Fluo-C3DH-A549-SIM data-set, since the manual annotations are available for all video frames, our VoxelEmbed achieved the best SEG and OP. The tracking performance is also competitive. For Fluo-N3DH-CHO and Fluo-N3DH-CE, only sparse manual annotations are provided (Fig. 4 (b)). As a result, the OP of Fluo-N3DH-CHO is inferior compared with leading approaches.

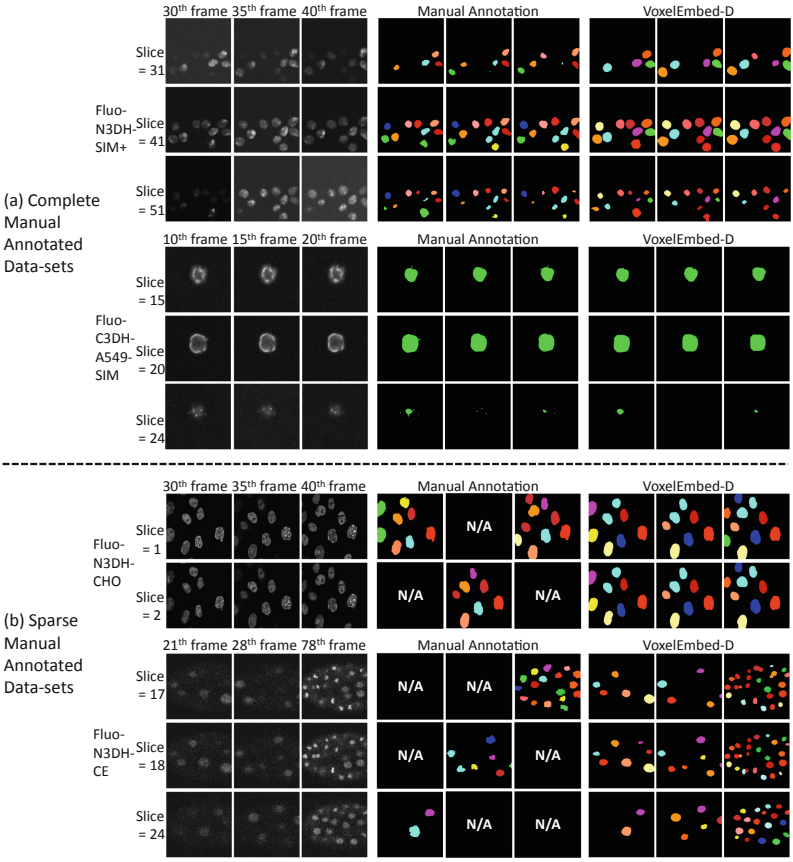

Fig. 4. The qualitative results of our VoxelEmbed framework. The 3D ISBI Cell Tracking Challenge dataset, (a) Complete and (b) Sparse manual annotations, as well as the VoxelEmbed results are presented.

5 Conclusion

In this paper, we introduce VoxelEmbed, an novel embedding based deep learning approach for 3D cell instance segmentation and tracking. The proposed detection method introduced a simple multi-stream training strategy to allow the embedding encoder to learn the spatial-temporal consistent voxel embedding across 3D context. The results show that the VoxelEmbed achieves decent performance compared with the leading method in four Cell Tracking Challenge datasets, with a uniformed learning framework. This study shows the promises of performing 3D cell instance segmentation and tracking with embedding based deep learning with a single GPU.

References

1. Appel, K., Haken, W., et al.: Every planar map is four colorable. Bull. Am. Math. Soc. **82**(5), 711–712 (1976)
2. Ballas, N., Yao, L., Pal, C., Courville, A.: Delving deeper into convolutional networks for learning video representations. arXiv preprint arXiv:1511.06432 (2015)
3. Cai, J., et al.: Accurate weakly-supervised deep lesion segmentation using large-scale clinical annotations: slice-propagated 3D mask generation from 2D RECIST. In: Frangi, A.F., Schnabel, J.A., Davatzikos, C., Alberola-López, C., Fichtinger, G. (eds.) MICCAI 2018. LNCS, vol. 11073, pp. 396–404. Springer, Cham (2018). https://doi.org/10.1007/978-3-030-00937-3_46
4. Cao, M., et al.: The single-cell transcriptional landscape of mammalian organogenesis. Nature **566**(7745), 496–502 (2019)
5. von Chamier, L., Laine, R.F., Henriques, R.: Artificial intelligence for microscopy: what you should know. Biochem. Soc. Trans. **47**(4), 1029–1040 (2019)
6. Chen, B.C., et al.: Lattice light-sheet microscopy: imaging molecules to embryos at high spatiotemporal resolution. Science **346**(6208), 1257998 (2014)
7. Condeelis, J., Pollard, J.W.: Macrophages: obligate partners for tumor cell migration, invasion, and metastasis. Cell **124**(2), 263–266 (2006)
8. Debeir, O., Van Ham, P., Kiss, R., Decaestecker, C.: Tracking of migrating cells under phase-contrast video microscopy with combined mean-shift processes. IEEE Trans. Med. imaging **24**(6), 697–711 (2005)
9. Meijering, E., Dzyubachyk, O., Smal, I.: Methods for cell and particle tracking. Methods Enzym. **504**, 183–200 (2012)
10. ISBI: Isbi cell tracking challenge benchmark leader boarder (2021). http://celltrackingchallenge.net/latest-ctb-results/
11. Jiang, C., Tsai, Y.J.: Enhanced crack segmentation algorithm using 3D pavement data. J. Comput. Civil Eng. **30**(3), 04015050 (2016)
12. Jiang, R., Gouvea, J., Hammer, D., Aeron, S.: Automatic coding of students' writing via contrastive representation learning in the wasserstein space. arXiv preprint arXiv:2011.13384 (2020)
13. Jin, B., Cruz, L., Goncalves, N.: Deep facial diagnosis: deep transfer learning from face recognition to facial diagnosis. IEEE Access **8**, 123649–123661 (2020)
14. Li, M., Chen, X., Li, X., Ma, B., Vitányi, P.M.: The similarity metric. IEEE Trans. Inform. Theory **50**(12), 3250–3264 (2004)
15. Liu, Q., et al.: Towards annotation-free instance segmentation and tracking with adversarial simulations. arXiv preprint arXiv:2101.00567 (2021)
16. Liu, T.L., et al.: Observing the cell in its native state: imaging subcellular dynamics in multicellular organisms. Science **360**(6386), eaaq1392 (2018)
17. Magnusson, K.E.: Segmentation and tracking of cells and particles in time-lapse microscopy. Ph.D. thesis, KTH Royal Institute of Technology (2016)
18. Matula, P., Maška, M., Sorokin, D.V., Matula, P., Ortiz-de Solórzano, C., Kozubek, M.: Cell tracking accuracy measurement based on comparison of acyclic oriented graphs. PloS ONE **10**(12), e0144959 (2015)
19. Meijering, E.: A bird's-eye view of deep learning in bioimage analysis. Comput. Struct. Biotech. J. **18**, 2312 (2020)
20. Microsoft: Azure NC-series (2020). https://docs.microsoft.com/en-us/azure/virtual-machines/nc-series
21. Newell, A., Yang, K., Deng, J.: Stacked hourglass networks for human pose estimation, pp. 483–499 (2016)

22. Niwattanakul, S., Singthongchai, J., Naenudorn, E., Wanapu, S.: Using of Jaccard coefficient for keywords similarity. In: Proceedings of the International Multiconference of Engineers and Computer Scientists, vol. 1, pp. 380–384 (2013)
23. NVIDIA: Nvidia, V. (2013). tesla k20 gpu accelerator board specification (2015). https://www.nvidia.com/content/PDF/kepler/tesla-k20-active-bd-06499-001-v03.pdf
24. Ong, E.Z., et al.: A dynamic immune response shapes Covid-19 progression. Cell Host Microbe **27**(6), 879–882 (2020)
25. Payer, C., Štern, D., Feiner, M., Bischof, H., Urschler, M.: Segmenting and tracking cell instances with cosine embeddings and recurrent hourglass networks. Med. Image Anal. **57**, 106–119 (2019)
26. Ulman, V., et al.: An objective comparison of cell-tracking algorithms. Nat. Methods **14**(12), 1141–1152 (2017)
27. Wan, Y., McDole, K., Keller, P.J.: Light-sheet microscopy and its potential for understanding developmental processes. Annu. Rev. Cell Dev. Biol. **35**, 655–681 (2019)
28. Yuan, W., Xu, W.: Neighborloss: a loss function considering spatial correlation for semantic segmentation of remote sensing image. IEEE Access **9**, 75641–75649 (2021)
29. Zhao, M., Chang, C.H., Xie, W., Xie, Z., Hu, J.: Cloud shape classification system based on multi-channel CNN and improved FDM. IEEE Access **8**, 44111–44124 (2020)
30. Zhao, M., et al.: Faster mean-shift: GPU-accelerated clustering for cosine embedding-based cell segmentation and tracking. Med. Image Anal. **71**, 102048 (2021)
31. Zhou, X., Wong, S.T.: High content cellular imaging for drug development. IEEE Signal Process. Mag. **23**(2), 170–174 (2006)