Pattern Recognition Letters 181 (2024) 37–42

Contents lists available at ScienceDirect

Pattern Recognition Letters

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

YOLO2U-Net: Detection-guided 3D instance segmentation for microscopy

Amirkoushyar Ziabari [a,*], Derek C. Rose [a], Abbas Shirinifard [b], David Solecki [b]

[a] Oak Ridge National Lab, 5200, 1 Bethel Valley Rd, Oak Ridge, 37830, TN, USA
[b] St. Jude Children's Hospital, Memphis, 37923, TN, USA

ARTICLE INFO

Editor: Jiwen Lu

Keywords:
Cell microscopy
3D instance segmentation
Deep learning

ABSTRACT

Microscopy imaging techniques are instrumental for characterization and analysis of biological structures. As these techniques typically render 3D visualization of cells by stacking 2D projections, issues such as out-of-plane excitation and low resolution in the z-axis may pose challenges (even for human experts) to detect individual cells in 3D volumes as these non-overlapping cells may appear as overlapping. A comprehensive method for accurate 3D instance segmentation of cells in the brain tissue is introduced here. The proposed method combines the 2D YOLO detection method with a multi-view fusion algorithm to construct a 3D localization of the cells. Next, the 3D bounding boxes along with the data volume are input to a 3D U-Net network that is designed to segment the primary cell in each 3D bounding box, and in turn, to carry out instance segmentation of cells in the entire volume. The promising performance of the proposed method is shown in comparison with current deep learning-based 3D instance segmentation methods.

1. Introduction

Advances in training deep convolutional neural networks have driven development in network architectures that are focused on semantic segmentation to pixel-wise label images. These approaches have largely focused on natural image segmentation (COCO [1] and PASCAL VOC [2]), though encoder–decoder networks with skip connections [3] and other multi-scale techniques (spatial pyramids [4] and atrous convolution or pooling [5]) have also shown success across a variety of medical and other imagery.

Biomedical image analysis presents challenges which are somewhat unique in instance segmentation. The orientation and concentration of objects can be random, objects can appear at varying scales, object boundaries can be unclear or overlapping, and object texture can vary spatially and contextually. Low contrast and noise or imaging artifacts such as out-of-plane excitation can make separating objects tedious and difficult to automate, often leading to a shortage of labeled data. Biomedical data sets are often inherently 3D, though potentially highly anisotropic with lower depth-wise resolution. This work is motivated by the segmentation of nuclei in 3D microscopy volumes. Typical image processing and machine learning based algorithms, e.g. [6,7], mostly focus on instance segmentation of cells in 2D and do not consider the aforementioned 3D challenges associated with volumetric data. Further, many recent works have extensively used Fully Convolutional Neural (FCNs) [8] to develop state-of-the-art instance segmentation algorithms [3,5,9–11] are mainly suitable for 2D instance segmentation of natural scenes; and/or are computationally expensive for 3D segmentation of a full image volume. It is emphasized that, in this work, 3D explicitly implies the *dimensionality* of an object in an image volume and not the *depth* of an object in a 2D image, such as in recent transformer-based 3D instance segmentation work [12].

A number of approaches have been taken to address computational limitations in 3D segmentation, spanning from integrating tri-planar views to recurrent neural networks for capturing slice to slice context [13]. 3D convolution with a U-Net topology for relatively small volumes (order of $100 \times 100 \times 100$ voxels) was performed successfully on biomedical imagery by Dou et al. [14] and Çiçek et al. [15]. Recently, Dunn et al. [16] developed DeepSynth, which combines SpCycleGAN, used to generate synthetic cell data for training, with a modified 3D U-Net network to 3D segment real cell data. DeepSynth encompasses a slice-by-slice based watershed and morphological post-processing algorithm for instance segmentation of touching cells. DeepCell [17,18] is another state-of-the-art method for instance image segmentation of volumes containing overlapping cells. DeepCell's deep learning-based watershed segmentation approach can handle overlapping cells in noisy volumes without over-segmentation, which is a typical drawback of traditional watershed algorithms. Another recent approach is StarDist that allows 3D segmentation of cells that are star-convex shaped objects [19]. For convex shapes and for small size

* Corresponding author.
E-mail addresses: ziabariak@ornl.gov (A. Ziabari), rosedc@ornl.gov (D.C. Rose), Abbas.Shirinifard@STJUDE.ORG (A. Shirinifard), David.Solecki@STJUDE.ORG (D. Solecki).

https://doi.org/10.1016/j.patrec.2024.03.015
Received 15 February 2023; Received in revised form 4 December 2023; Accepted 18 March 2024
Available online 24 March 2024

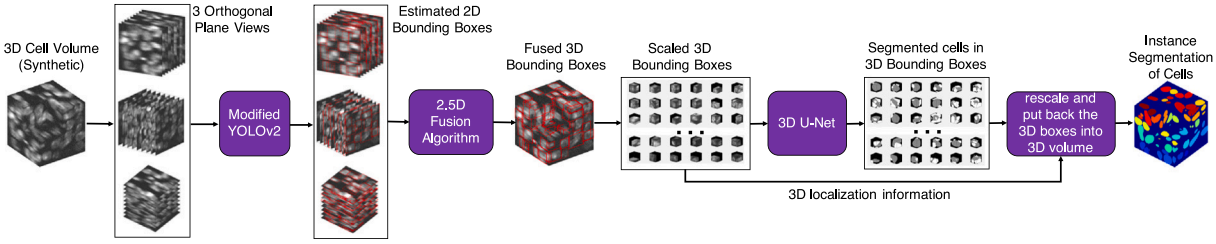

Fig. 1. YOLO2U-Net architecture.

volumes, StarDist can be a very powerful tool [20]. Other recent works [21–23], have modified and leveraged Mask R-CNN [24] to perform instance segmentation in various 2D images, in particular in the context of 2D cell microscopy images [21], but are not readily extended to 3D.

In this paper, a comprehensive detection and segmentation framework called YOLO2U-Net is presented. YOLO2U-Net judiciously combines two successful network topologies, namely You-Only-Look-Once (YOLO) [25] and U-Net [15], to do 3D object detection and instance segmentation for cells in microscopy volumes. In the proposed method, YOLO [25] is first modified and used to detect cells from 2D orthogonal perspectives of a 3D volume and subsequently extract a bounding box around each individual cell. Following detection, an algorithm that combines the 2D detected bounding boxes localizes the cells in 3D bounding boxes. Finally, instance-level segmentation of 3D cells within the detected 3D bounding boxes is performed using a 3D U-Net [15] modified for unbalanced data. This 3D network segments out the primary cells in each bounding cube. The proposed method is an extension to the authors' work in [26]. In comparison the proposed method removes the necessity of any post-processing watershed and morphological operations for separating the cells; rather the entire cell will be segmented out in each 3D bounding box. This in turn avoids common artifacts when performing instance segmentation associated with stitching sub-volumes together such as over-segmentation of cells and missing cell–cell boundaries (especially for high cell confluence cases).

Section 2 describes the proposed method and its comprising components. Section 3 include synthetically generated data along with metrics, experimental and quantitative comparisons with the state-of-the-art methods. The paper concludes and future plans are discussed in Section 4.

2. YOLO2U-Net

Fig. 1 summarizes the proposed method. This method first localizes the cells using a 2.5D fusion algorithm that fuses the 2D bounding boxes of the cells that are obtained from 2D orthogonal perspectives of the 3D volume of cells. The YOLO [25] network obtains 2D bounding boxes in each perspective. The volume of data with localization information are then input to 3D U-Net [15], which is trained to identify the main cells in each 3D bounding box and separate them from portions of the neighboring cells within the same box. Finally, YOLO2U-Net leverages the knowledge about the position of individual boxes to compose the cells back into the original volume. The following describes each component of the proposed method separately.

2.1. 2.5D YOLO-based fusion algorithm

YOLO, and its more recent version YOLOv2 [25], is a state-of-the-art fast 2D object detection algorithm. Despite its utility for fast localization in 2D, it becomes exponentially more expensive to perform 3D localization with 3D convolutions within the YOLOv2 topology. To leverage the performance of YOLOv2 for 3D localization of objects in 3D volumes of data the 2.5D fusion algorithm outlined in Algorithm 1 is proposed.

In this algorithm the YOLOv2 network is trained on 2D slices from orthogonal Cartesian planes of a 3D synthetic image volume. YOLOv2 may produce several bounding boxes per object; only bounding boxes with more than 50% confidence are kept. This step is then followed by a non-maximum suppression to discard boxes that have more than 50% overlap to make sure each object is localized only once. For each object that extends in 3D, there must be a nonempty region of intersection between bounding boxes in each view. Therefore, all the detected 2D bounding boxes are pairwise compared with other boxes in the same plane as well as with boxes from orthogonal planes. The coordinates of the overlapped 2D boxes are joined to obtain the proposal coordinates for the 3D bounding boxes ($[x_{min}, x_{max}, y_{min}, y_{max}, z_{min}, z_{max}]$). Next, to prune the multiple boxes created from multi-perspective detection, all the proposal 3D bounding boxes are pair-wise compared and those with more than 5% overlap are clustered together. This threshold was empirically chosen by testing the fusion algorithm on synthetic volumes of touching spheres. Finally, to obtain the final 3D bounding boxes' coordinates, for each cluster the median of each of the 6 coordinates of all the extracted 3D bounding boxes are calculated and a non-maximal suppression is applied.

2.2. 3D U-Net for 3D cell segmentation inside 3D bounding boxes

Since the cells are localized, their positional information can be used to perform guided bounding box selection for 3D instance segmentation with a 3D U-Net. Here, the U-Net is trained to separate the main cell from the remaining voxels in the 3D bounding box. To mitigate for cells of differing shape and size, during training 3D input cubes of cells are scaled to a fixed size of ($48 \times 48 \times 48$). This size is chosen based on the cell sizes encountered in the training data and for faster training of the 3D U-Net. For example, if a bounding box is of size of $59 \times 30 \times 47$, it is zero-padded to make it $59 \times 59 \times 59$ and then scaled to $48 \times 48 \times 48$. Once the segmentation inside the bounding box is performed it is resized back to its original size. The proposed one-step instance segmentation strategy avoids the post-processing watershed and morphological filtering such as those in [16,26]. After all the bounding boxes are segmented, they are placed in separate volumes of the size of the original input data at their position. An *argmax* is then applied to this 4D volume to label cells for the final 3D volume.

3. Experimental results

3.1. Data sets

CompuCell3D is an open-source toolkit that is widely used to simulate biological cells and tissues [27] using agent-based methods. A three-compartment virtual cell was used to simulate cell nucleus shapes and internal distribution of DNA material. These virtual cells are flexible and can easily mimic realistic cell nucleus shape and cell–cell boundaries. Euchromatin domains are modeled as two equally sized compartments occupying about 85% of the total volume of a virtual cell. Heterochromatin domains are modeled as multiple compartments

Algorithm 1 2.5D YOLO-Based Fusion Algorithm and Sub-Algorithms

Main Algorithm:
Input: 2D images in XY, XZ, YZ views
Output: List of 3D bounding boxes
$\mathbf{B} \leftarrow \{\}$ ▷ Initialize an empty set for 3D bounding boxes
for $view \in \{XY, XZ, YZ\}$ **do**
  $I \leftarrow$ Image in view
  $\mathbf{preds}_{view} \leftarrow \{\}$ ▷ Initialize an empty list for predictions in each view
  for $\mathbf{p} \in YOLOv2(I)$ **do** ▷ Predict bounding boxes using YOLOv2
    if $\mathbf{p}$.confidence > 0.5 **then**
      $\mathbf{preds}_{view} \leftarrow \mathbf{preds}_{view} \cup \{\mathbf{p}\}$ ▷ Add high confidence predictions
    end if
  end for
end for
$\mathbf{B} \leftarrow$ PairwiseCompare($\mathbf{preds}_{XY}, \mathbf{preds}_{XZ}, \mathbf{preds}_{YZ}$)
$\mathbf{B} \leftarrow$ Estimate3DCoordinates($\mathbf{B}$)
$\mathbf{B} \leftarrow$ ClusterBoundingBoxes($\mathbf{B}, 0.05$)
$\mathbf{B} \leftarrow$ MedianClusterMerge($\mathbf{B}$)
$\mathbf{B} \leftarrow$ Non-maximalSuppression($\mathbf{B}$)
return B
Sub-Algorithm 1: PairwiseCompare Function
Input: Bounding box predictions $\mathbf{preds}_{XY}, \mathbf{preds}_{XZ}, \mathbf{preds}_{YZ}$
Output: Grouped bounding boxes
$\mathbf{G} \leftarrow \{\}$ ▷ Initialize an empty set for grouped boxes
for each bounding box $\mathbf{b}_{XY}$ in $\mathbf{preds}_{XY}$ **do**
  for each bounding box $\mathbf{b}_{XZ}$ in $\mathbf{preds}_{XZ}$ **do**
    for each bounding box $\mathbf{b}_{YZ}$ in $\mathbf{preds}_{YZ}$ **do**
      if $\mathbf{b}_{XY}, \mathbf{b}_{XZ}, \mathbf{b}_{YZ}$ overlap **then**
        $\mathbf{G} \leftarrow \mathbf{G} \cup \{\mathbf{b}_{XY}, \mathbf{b}_{XZ}, \mathbf{b}_{YZ}\}$
      end if
    end for
  end for
end for
return G
Sub-Algorithm 2: Estimate3DCoordinates Function
Input: Grouped bounding boxes $\mathbf{G}$
Output: 3D bounding boxes
$\mathbf{B_{3D}} \leftarrow \{\}$
for each group g in $\mathbf{G}$ **do**
  Initialize $x_{min,est}, x_{max,est}, y_{min,est}, y_{max,est}, z_{min,est}, z_{max,est}$
  for each bounding box $\mathbf{b}$ in g **do**
    Update $x_{min,est}, x_{max,est}, y_{min,est}, y_{max,est}, z_{min,est}, z_{max,est}$ based on $\mathbf{b}$
  end for
  Add $[x_{min,est}, y_{min,est}, z_{min,est}, x_{max,est}, y_{max,est}, z_{max,est}]$ to $\mathbf{B_{3D}}$
end for
return $\mathbf{B_{3D}}$
Sub-Algorithm 3: ClusterBoundingBoxes Function
Input: 3D bounding boxes $\mathbf{B_{3D}}$
Output: Clustered bounding boxes
$\mathbf{C} \leftarrow \{\}$ ▷ Initialize an empty set for clusters
for each bounding box $\mathbf{b}$ in $\mathbf{B_{3D}}$ **do**
  Find all boxes in $\mathbf{B_{3D}}$ with more than 5% overlap with $\mathbf{b}$
  Form a cluster with these boxes and add to $\mathbf{C}$
end for
return C
Sub-Algorithm 4: MedianClusterMerge Function
Input: Clusters of bounding boxes $\mathbf{C}$
Output: Merged bounding boxes
$\mathbf{M} \leftarrow \{\}$ ▷ Initialize an empty set for merged boxes
for each cluster c in $\mathbf{C}$ **do**
  Initialize six sets $S_{x_{min}}, S_{x_{max}}, S_{y_{min}}, S_{y_{max}}, S_{z_{min}}, S_{z_{max}}$
  for each bounding box $\mathbf{b}$ in c **do**
    Add each coordinate of $\mathbf{b}$ to its respective set
  end for
  Calculate the median of each set
  Form the merged bounding box using these medians
  Add the merged box to $\mathbf{M}$
end for
return M
Sub-Algorithm 5: Non-maximalSuppression Function
Input: Merged bounding boxes $\mathbf{M}$
Output: Final set of bounding boxes
$\mathbf{F} \leftarrow \{\}$ ▷ Initialize an empty set for final boxes
for each bounding box $\mathbf{b}$ in $\mathbf{M}$ **do**
  Find all boxes in $\mathbf{M}$ with more than 50% overlap with $\mathbf{b}$
  Choose the box with the highest confidence score
  Add the chosen box to $\mathbf{F}$
end for
return F

(5 to 9) occupying about 15% of the total volume. 128 cells were randomly initialized located in a 84 × 84 × 84 lattice. The lattice was cropped to 64 × 64 × 64 to avoid artifacts due to lattice boundary conditions. The extent of cell–cell contact and organization of the internal compartments can be adjusted by setting appropriate contact energy parameters. To increase cell–cell contact, a negative surface tension is required (and vice versa). The 3-compartment virtual cells in the 64 × 64 × 64 lattice volume are transformed to a realistic synthetic image by: (1) up-scaling and creating smooth cell boundary masks and (2) assigning signal intensity to compartments and applying smooth boundary masks. To mimic real microscopy data sets, realistic microscopy aberrations (experimental point spread function approximated by a 3D Gaussian Blur) are applied and added with Gaussian noise to create the final simulated data.

3.2. Metrics

In this section, the metrics that are used throughout the main text are summarized. The intersection-over-union (IoU) between two 3D segmented cells is defined as $IoU = \frac{Cell_{target} \cap Cell_{predicted}}{Cell_{target} \cup Cell_{predicted}}$, where, $Cell_{target}$ and $Cell_{predicted}$ correspond to voxels of the cell target cell and predicted cell in the 3D volume. To evaluate the instance segmentation performance, precision (P), recall (R), and Jaccard (J) scores are used.

Precision (P) measures the accuracy of the positive predictions. In the context of 3D cell segmentation, a high precision indicates that most of the voxels labeled as part of a cell truly belong to the cell, minimizing false positives. Recall assesses the ability of the model to identify all relevant instances. In this case, a high recall means the algorithm successfully identifies most of the cell voxels, reducing false negatives. Jaccard (J) Score is a measure of the overlap between the predicted and actual labels. For 3D cell segmentation, a high Jaccard score indicates that the predicted segmentation closely matches the true cell shapes and boundaries.

These metrics are calculated at the voxel level and as a function of 3D IOU threshold values (th), using:

$$\begin{aligned} P(th) &= \frac{N_{TP}(th)}{N_{TP}(th) + N_{FP}(th)}, \\ R(th) &= \frac{N_{TP}(th)}{N_{TP}(th) + N_{FN}(th)}, \\ J(th) &= \frac{N_{TP}(th)}{N_{TP}(th) + N_{FP}(th) + N_{FN}(th)}. \end{aligned} \tag{1}$$

Here, N is the number of voxels and TP, FP, and FN are true positive count, true negative count, and false negative count. The average values for N_{vols} test volumes are calculated at each 3D IOU threshold level as average precision AP, average recall AR and average Jaccard AJ scores. By integrating these values over the entire range of IOU levels, the mean average precision (mAP), mean average recall (mAR), and mean average Jaccard (mAJ) score are obtained.

3.3. Comparison with other methods

A comparison of YOLO2U-Net is made against three existing methods, namely, DeepCell [17], DeepSynth [16], and a two-tier CNN [26] previously presented by the authors. To perform a fair comparison, all methods are trained on the same training data sets, which is 20 volumes of 128^3 simulated cell microscopy data as detailed in Section 3.1.

The trained network, then tested on 20 new volumes of cell data that are the same size as training volumes but of course not seen in the training. An example of test volumes along with the ground truth instance segmentation mask are shown in Fig. 2. Fig. 3, compares precision, recall, and the Jaccard scores obtained by different methods for test data sets. Each panel in this Figure plots an average score for 20 volumes in the test data set as a function of 3D intersection-over-union (IoU) levels (in range $\in [0.5, 1]$). Note that YOLO2U-Net performs best among the tested methods in all cases for IOU larger than 0.7 and has about the same performance as DeepSynth at lower IOUs. The improved performance of YOLO2U-Net at higher IOUs is indicative

Table 1
Metric comparison.

	DeepCell [17]	DeepSynth [16]	Two-Tier CNN [26]	**YOLO2U-Net**
mAP	0.16	0.338	0.296	**0.367**
mAR	0.187	0.349	0.331	**0.39**
mAJ	0.107	0.248	0.211	**0.263**

of the importance of the 3D localization performed prior to instance segmentation.

To further demonstrate the impact of 3D localization, Fig. 4 shows slices of segmented cells in a test volume from different views in . For these examples, and in particular in X-Z and Y-Z slices, where blurring and out-of-plane excitation worsens the image quality, DeepCell fails to accurately segment boundaries, DeepSynth suffers from over-segmentation due to post-processing watershed, two-tier CNN misses some cells, while YOLO2U-Net correctly separates cells even when the boundaries are vague and very blurred. These observations, again, signify the importance of the localization step. Note that the slices in Fig. 4 are intentionally selected to contain notable cell configurations that highlight both the strengths and limitations of the YOLO2U-Net approach.

Table 1 compares the three methods in terms of their mean average precision (mAP), mean average recall (mAR), and mean average Jaccard (mAJ) scores. The largest values are shown in bold. In all cases, YOLO2U-Net outperforms other tested methods.

3.4. Ablation study

An ablation study was conducted to investigate the impact and contribution of each of the components of the proposed architecture (2D localization, detection box fusion for 3D bounding boxes, and 3D segmentation) on the performance of YOLO2U-Net.

First, consider the case that the 3D bounding boxes are perfectly known — Baseline 1 (3DGTBBs). This is equivalent to a case in which YOLOv2 and Algorithm 1 both have perfect performance. Baseline 1 aims to show what the 3D U-Net can achieve using perfect inputs. Second, assume that only 2D bounding boxes of the cells are known — Baseline 2 (2DGTBBs). Algorithm 1 performs fusion of perfect 2D boxes and obtains 3D bounding boxes of the cells before inputting the data into 3D U-Net. Baseline 2 evaluates the impact of the fusion approach and algorithm. These baselines are used to directly compare with the full proposed YOLO2U-Net method as shown in Fig. 5.

Baselines are compared to the full method for the three data sets and the average metric score is plotted as a function of IoU for each data set. The three data sets tested are of increasing complexity. In data set 1, noise is added to simulated CompuCell3D data; in 2, noise and Gaussian blur are both added; and in (3) realistic microscopy aberrations and noise are added (same data as in the previous sections). It is evident from the figure that, for the less challenging cases in data set 1 and 2, a near perfect score is obtainable by improving YOLOv2 (or using alternative methods) and the fusion algorithm. Further, enhancing just the fusion algorithm for these data sets will improve the cell counting accuracy (Jaccard score) for YOLO2U-Net. For data set 3, even using perfect bounding boxes does not help in distinguishing and segmenting all cells correctly. This observation suggests that improvements to the current 3D U-Net are necessary for better instance segmentation of realistic data sets. Table 2 summarizes the mAJ values for the three scenarios discussed. This study clarifies that an improvement to YOLOv2 or replacement with a better 2D detection method can lead to significantly better performance with YOLO2U-Net. The same argument is valid for the fusion algorithm.

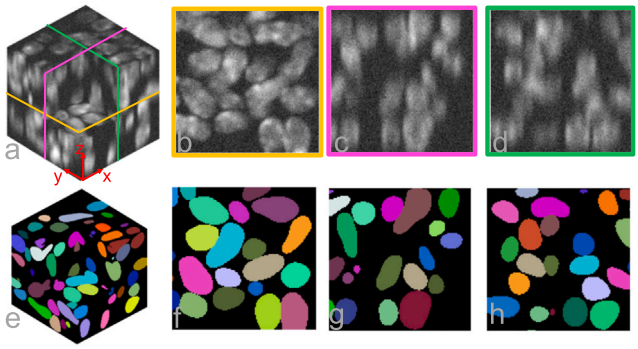

Fig. 2. An example volume from test data set is shown along with the ground truth instance segmentation. Three central cross sections along axial (XY), coronal (XZ), and sagittal (XZ) planes are also shown.

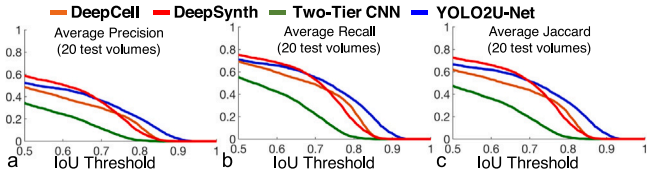

Fig. 3. Quantitative comparison: (AP),(AR), (AJ) scores (defined in the text) for 20 volumes against IoU are plotted.

Table 2
Ablation study of mAJ reliance on architecture components.

	Baseline 1 (3DGTBBs)	Baseline 2 (2DGTBBs)	YOLO2U-Net
Data set1	0.923	0.647	0.508
Data set2	0.798	0.567	0.442
Data set3	0.456	0.333	0.263

4. Conclusions and future work

In this work a novel, versatile, and modular neural network architecture, titled YOLO2U-Net, is proposed. This architecture combines two widely used deep learning architectures through an image processing-based fusion algorithm and performs joint detection, localization, and 3D instance segmentation of cell nuclei. The proposed method is (a) efficient by localizing segmentation computation, (b) adaptive to changes in object size through input re-scaling, and (c) modular to enable plug-and-play future-proofing. Several volumes of instance-level labeled data sets are simulated. These data sets challenge 3D instance segmentation models the same way real data does in two major aspects: (a) nontrivial cell geometry and cell–cell boundaries; and, (b) out-of-plane signal mixing and low in-plane resolution. This data will be made publicly available. After extensive searching, publicly available data sets containing characteristics of microscopy artifacts along with accurate instance segmentation masks were not previously found. The proposed method along with three 3D segmentation methods were trained and tested on the generated data sets. In all cases, YOLO2U-Net outperform these current methods. An ablation study was used to analyze the impact of different components of the network on its performance. Given the reported findings, integrating the components of the method into a model with full gradient path for end-to-end training to improve instance segmentation performance is currently being investigated. Further, to improve performance on the most challenging data, hyperparameter optimization techniques for the networks tested, using more recent versions of YOLO architecture (YOLOv8) [28], and dropping-in improved networks as modular replacements are being considered. One such example of the latter is replacing the 3D U-Net with recent vision-transformer based approaches such as in Swin-Unet [29] and Swin Unet3D [30]. Future work

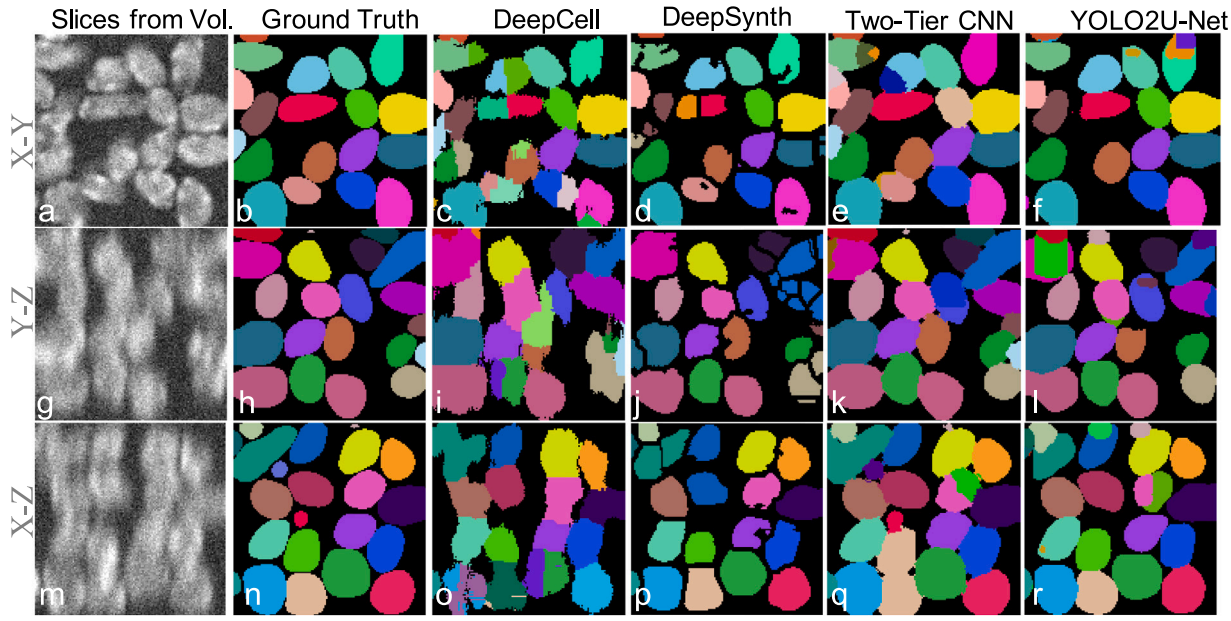

Fig. 4. Instance segmentation comparison across slices from different views for a volume from test data sets. Challenging slices are selected to highlight strengths and limitations of the YOLO2U-Net.

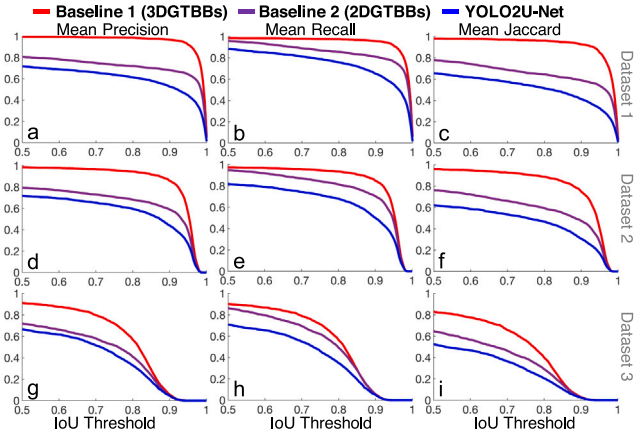

Fig. 5. Ablation study metrics to analyze the impact of YOLOv2 and the proposed fusion Algorithm 1 on the performance of YOLO2U-Net. Baseline 1: using perfect 3D bounding boxes in last stage (3D U-Net). Baseline 2: using perfect 2D bounding boxes (impact of box fusion). Mean precision (a, d, g), recall (b, e, h), and Jaccard scores (c, f, i) for 20 volumes against IoU are plotted.

will expand this approach to real data sets from different modalities and, when needed, apply GANs and domain adaptation to improve the performance.

CRediT authorship contribution statement

Amirkoushyar Ziabari: Conceptualization, Data curation, Formal analysis, Investigation, Methodology, Validation, Visualization, Writing – original draft, Writing – review & editing. **Derek C. Rose:** Investigation, Visualization, Writing – original draft, Writing – review & editing. **Abbas Shirinifard:** Data curation, Investigation, Writing – original draft. **David Solecki:** Funding acquisition, Project administration, Supervision.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Data availability

Data will be made available on request.

Acknowledgments

Research sponsored by the U.S. Department of Energy, under contract DE-AC05-00OR22725 with UT-Battelle, LLC. The US government retains and the publisher, by accepting the article for publication, acknowledges that the US government retains a nonexclusive, paid-up, irrevocable, worldwide license to publish or reproduce the published form of this manuscript, or allow others to do so, for US government purposes. DOE will provide public access to these results of federally sponsored research in accordance with the DOE Public Access Plan (http://energy.gov/downloads/doe-public-access-plan). This collaborations was funded by St. Jude Children's Research Hospital through funding from the American Lebanese Syrian Associated Charities (ALSAC). The Solecki Laboratory is funded by grants 1R01NS066936 and R01NS104029-02 from the National Institute of Neurological Disorders (NINDS).

References

[1] T.-Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, C.L. Zitnick, Microsoft COCO: Common Objects in Context, in: D. Fleet, T. Pajdla, B. Schiele, T. Tuytelaars (Eds.), Computer Vision – ECCV 2014, Springer International Publishing, 2014, pp. 740–755.

[2] M. Everingham, S.M. Eslami, L. Van Gool, C.K. Williams, J. Winn, A. Zisserman, The pascal visual object classes challenge: A retrospective, Int. J. Comput. Vis. 111 (1) (2014) 98–136.

[3] O. Ronneberger, P. Fischer, T. Brox, U-Net: Convolutional networks for biomedical image segmentation, in: Medical Image Computing and Computer-Assisted Intervention – MICCAI 2015, Springer International Publishing, Cham, 2015, pp. 234–241.

[4] H. Zhao, J. Shi, X. Qi, X. Wang, J. Jia, Pyramid scene parsing network, in: 2017 IEEE Conference on Computer Vision and Pattern Recognition, CVPR, IEEE, Honolulu, HI, 2017, pp. 6230–6239.
[5] L.-C. Chen, G. Papandreou, F. Schroff, H. Adam, Rethinking atrous convolution for semantic image segmentation, 2017, arXiv:1706.05587 [cs].
[6] K. Al-Dulaimi, V. Chandran, K. Nguyen, J. Banks, I. Tomeo-Reyes, Benchmarking HEp-2 specimen cells classification using linear discriminant analysis on higher order spectra features of cell shape, Pattern Recognit. Lett. 125 (2019) 534–541, http://dx.doi.org/10.1016/j.patrec.2019.06.020.
[7] P. Quelhas, M. Marcuzzo, A.M. Mendonca, A. Campilho, Cell nuclei and cytoplasm joint segmentation using the sliding band filter, IEEE Trans. Med. Imaging 29 (8) (2010) 1463–1473.
[8] J. Long, E. Shelhamer, T. Darrell, Fully convolutional networks for semantic segmentation, in: Proceedings of the IEEE Conference on Computer vision and Pattern recognition, 2015, pp. 3431–3440.
[9] K. He, G. Gkioxari, P. Dollár, R. Girshick, Mask R-CNN, in: The IEEE International Conference on Computer Vision, ICCV, 2017, arXiv:1703.06870.
[10] A.O. Vuola, S.U. Akram, J. Kannala, Mask-RCNN and U-Net ensemble for nuclei segmentation, in: 2019 IEEE 16th International Symposium on Biomedical Imaging, ISBI 2019, 2019, pp. 208–212.
[11] Q.-L. Zhang, Y.-B. Yang, A boundary-preserving conditional convolution network for instance segmentation, Pattern Recognit. Lett. 163 (2022) 1–9, http://dx.doi.org/10.1016/j.patrec.2022.09.003.
[12] J. Schult, F. Engelmann, A. Hermans, O. Litany, S. Tang, B. Leibe, Mask3D: Mask transformer for 3D semantic instance segmentation, in: 2023 IEEE International Conference on Robotics and Automation, ICRA, 2023, pp. 8216–8223, http://dx.doi.org/10.1109/ICRA48891.2023.10160590.
[13] J. Chen, L. Yang, Y. Zhang, M. Alber, D.Z. Chen, Combining fully convolutional and recurrent neural networks for 3D biomedical image segmentation, in: Advances in Neural Information Processing Systems, 2016, pp. 3036–3044.
[14] Q. Dou, H. Chen, Y. Jin, L. Yu, J. Qin, P.-A. Heng, 3D deeply supervised network for automatic liver segmentation from CT volumes, in: Medical Image Computing and Computer-Assisted Intervention – MICCAI 2016, Vol. 9901, Springer International Publishing, Cham, 2016, pp. 149–157.
[15] Ö. Çiçek, A. Abdulkadir, S.S. Lienkamp, T. Brox, O. Ronneberger, 3D U-net: Learning dense volumetric segmentation from sparse annotation, in: Medical Image Computing and Computer-Assisted Intervention – MICCAI 2016, Vol. 9901, Springer International Publishing, Cham, 2016, pp. 424–432.
[16] K.W. Dunn, C. Fu, D.J. Ho, S. Lee, S. Han, P. Salama, E.J. Delp, DeepSynth: Three-dimensional nuclear segmentation of biological images using neural networks trained with synthetic data, Sci. Rep. 9 (1) (2019) 1–15.
[17] M. Bai, R. Urtasun, Deep watershed transform for instance segmentation, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2017, pp. 5221–5229.
[18] W. Wang, D.A. Taft, Y.-J. Chen, J. Zhang, C.T. Wallace, M. Xu, S.C. Watkins, J. Xing, Learn to segment single cells with deep distance estimator and deep cell detector, Comput. Biol. Med. 108 (2019) 133–141.
[19] M. Weigert, U. Schmidt, R. Haase, K. Sugawara, G. Myers, Star-convex polyhedra for 3d object detection and segmentation in microscopy, in: The IEEE Winter Conference on Applications of Computer Vision, 2020, pp. 3666–3673.
[20] M. Stevens, A. Nanou, L.W.M.M. Terstappen, C. Driemel, N.H. Stoecklein, F.A.W. Coumans, StarDist image segmentation improves circulating tumor cell detection, Cancers 14 (12) (2022) http://dx.doi.org/10.3390/cancers14122916.
[21] A.R. Revanda, C. Fatichah, N. Suciati, Classification of Acute Lymphoblastic Leukemia on White Blood Cell Microscopy Images Based on Instance Segmentation Using Mask R-CNN, Vol. 15, 2022, pp. 625–637.
[22] K. Lv, Y. Zhang, Y. Yu, H. Wang, L. Li, H. Jiang, C. Dai, Contour deformation network for instance segmentation, Pattern Recognit. Lett. 159 (2022) 213–219, http://dx.doi.org/10.1016/j.patrec.2022.05.025.
[23] B.R. Kang, H. Lee, K. Park, H. Ryu, H.Y. Kim, BshapeNet: Object detection and instance segmentation with bounding shape masks, Pattern Recognit. Lett. 131 (2020) 449–455, http://dx.doi.org/10.1016/j.patrec.2020.01.024.
[24] K. He, G. Gkioxari, P. Dollár, R. Girshick, Mask R-CNN, in: 2017 IEEE International Conference on Computer Vision, ICCV, 2017, pp. 2980–2988, http://dx.doi.org/10.1109/ICCV.2017.322.
[25] J. Redmon, A. Farhadi, YOLO9000: Better, faster, stronger, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2017, pp. 7263–7271.
[26] A. Ziabari, A. Shirinifard, M. Eicholtz, D.J. Solecki, D.C. Rose, A Two-Tier Convolutional Neural Network for Combined Detection and Segmentation in Biological Imagery, in: 2019 IEEE Global Conference on Signal and Information Processing, GlobalSIP, IEEE, 2019, pp. 1–5.
[27] M.H. Swat, G.L. Thomas, J.M. Belmonte, A. Shirinifard, D. Hmeljak, J.A. Glazier, Multi-scale modeling of tissues using CompuCell3D, in: Methods in Cell Biology, Vol. 110, Elsevier, 2012, pp. 325–366.
[28] Ultralytics, YOLOv8: An implementation by ultralytics, 2023, Available from: https://github.com/ultralytics/ultralytics.
[29] H. Cao, Y. Wang, J. Chen, D. Jiang, X. Zhang, Q. Tian, M. Wang, Swin-Unet: Unet-like pure transformer for medical image segmentation, in: L. Karlinsky, T. Michaeli, K. Nishino (Eds.), Computer Vision – ECCV 2022 Workshops, Springer Nature Switzerland, Cham, 2023, pp. 205–218.
[30] Y. Cai, Y. Long, Z. Han, M. Liu, Y. Zheng, W. Yang, L. Chen, Swin Unet3D: a three-dimensional medical image segmentation network combining vision transformer and convolution, BMC Med. Inform. Decis. Mak. 23 (1) (2023) 33, http://dx.doi.org/10.1186/s12911-023-02129-z.