FAST PARTICLE PICKING FOR CRYO-ELECTRON TOMOGRAPHY USING ONE-STAGE DETECTION

Shiyu Wu[1,2]*, Guole Liu*[1,2]*, Ge Yang*[1,2]
[1]National Laboratory of Pattern Recognition, Institute of Automation, Chinese Academy of Sciences, Beijing, 100190, China
[2]School of Artificial Intelligence, University of Chinese Academy of Sciences, Beijing, 100049, China

ABSTRACT

Cryo-electron tomography (Cryo-ET) is an electron microscopy technique that plays an important role in structural biology by reconstructing structures of biological macromolecules in their native environment. In cryo-ET images, also called tomograms, macromolecules are detected through particle picking for their structural reconstruction. Automated particle picking is essential for processing large volumes of cryo-ET data. Although deep learning-based object detection models have achieved excellent performance in many applications, their adoption in particle picking for tomograms remains limited due to low signal-to-noise ratios (SNRs) of cryo-ET images, typically below 0.1. So far, studies on particle picking techniques for tomograms have chosen segmentation models for accuracy. Different from these studies, we solve the problem as a 3D object detection task. Specifically, we have developed a one-stage detection model that locates and classifies particles in 3D tomograms with high efficiency and competitive accuracy. Unlike segmentation models, our model requires only location and class information of particles but not their geometry information for training. Experiments show that our model achieves detection accuracy similar as that of state-of-the-art segmentation models on the SHREC2020 dataset of synthetic images. But its detection speed is about ten times faster than the fastest segmentation model. Our model also achieves good performance on the EMPIAR-10045 dataset of real cryo-ET images. Source code and data of this work are openly accessible at: http://github.com/cbmi-group/3DFastParticleDetection.

Index Terms**—**Particle picking, Cryo-electron tomography, Object detection, Deep learning, Convolutional neural network

1. INTRODUCTION

Cryo-electron tomography (Cryo-ET) makes it possible to solve three-dimensional structures of biological macromolecules in their native environment [1]. In cryo-ET, quick-frozen samples of biomacromolecules are transmission scanned from multiple angles. Then the acquired series of 2D projections are used to reconstruct 3D tomograms [2]. However, biological samples can only withstand limited radiation, which results in very low signal-to-noise ratios (SNRs) in tomograms, typically below 0.1. This makes it infeasible to reconstruct structures of biomacromolecules in 3D tomograms directly. Instead, thousands or more images of biomacromolecules of the same kind must be aligned and averaged to reconstruct structures at nanometer to angstrom resolutions [1]. A key requirement of this procedure is to pick as many biomacromolecules, also called particles, from the tomograms as possible. As manual picking is laborious and inefficient to handle large volumes of cryo-ET data, automated particles picking becomes essential.

Over the past two decades, a wide variety of automated particle picking techniques have been proposed [3]. However, these techniques are mainly developed to pick single particles from 2D cryo-electron microscopy (cryo-EM) images rather than 3D tomograms. Before the rapid rise of deep learning, a commonly used approach is template matching [4-7]. However, it can only handle biomacromolecules with known structures. And its performance is influenced strongly by the quality of the templates prepared. Another approach is to use edge detection type algorithms such as Difference of Gaussian [8] and Laplacian of Gaussian [9]. These algorithms are template-free and can detect particles of unknown structures with fast speeds. But they perform well only on cryo-EM images with relatively high SNRs [8].

Deep learning brings a new solution to the particle picking problem. Recently, significant progress has been made in object detection using deep learning, with many models developed [10]. One-stage models such as YOLO [11] and SSD [12] combine object localization and classification into one step, while two-stage models such as Fast R-CNN [13], Faster R-CNN [14], and Mask R-CNN [15] complete object localization and classification in two steps. These models have also been used to pick single particles from 2D cryo-EM images. For example, Wang *et al.* [16] proposed DeepPicker, a convolutional neural network (CNN) model that detects particles in an automated fashion with accuracy approaching that of manual annotation. Wagner *et al.* [17] developed SPHIRE-crYOLO, a one-stage particle picking model based on YOLO [11] for fast speeds in selecting single particles. Li *et al.* [18] proposed a model based on Faster R-CNN for high accuracy and robustness in particle detection and classification.

For 3D tomograms, motived by the success of U-Net [19, 20] in segmentation of biomedical images, a natural way for particle picking is to separate pixels of particles from their background and then cluster them to obtain the location of each particle. An example of this approach is DeepFinder [21], which utilizes a U-Net architecture to identify multi-class macromolecules. Other studies [2, 22] use different

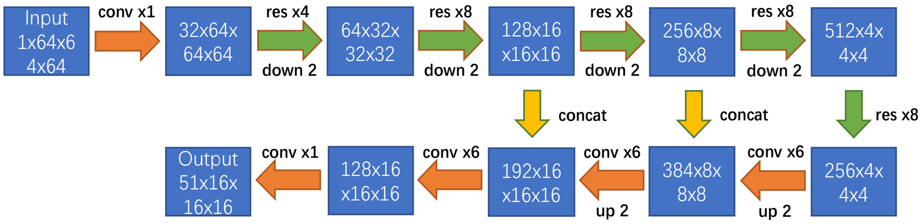

Figure 1. Architecture of the proposed particle detection model. “res” denotes a ResNet block that contains two convolutional blocks and a skip connection. “Up” and “down” denote upsampling and downsampling, respectively.

segmentation models to achieve comparable performance. Overall, these models utilize geometry information of particles to achieve high detection accuracy. However, they also have two important limitations. First, they require 3D segmentation labels of various kinds for training. Second, their clustering requires long processing time. To overcome these limitations, we choose to solve the problem of particle picking for 3D tomograms as an object detection task.

Our main research contributions are as follows: (1) We have developed a one-stage object detection model for automated particle picking from 3D tomograms. It generally matches state-of-the-art segmentation models in detection accuracy but is much faster in detection speed. (2) We have developed a strategy to handle large tomograms. It divides a large tomogram into small overlapping cubes so that detection performance will not be negatively affected by those particles that cross cube boundaries. This strategy is highly effective and can also be combined with other DNNs models for particle picking.

2. METHOD

2.1. Data Preprocessing: Dividing Tomograms into Overlapping Cubes

Sizes of 3D cryo-ET tomograms are often much larger than those of 2D cryo-EM images, which cause deep learning models to vastly exceed the memory limit of GPUs. It is essential to divide tomograms into small pieces before training. A simple strategy is to split a tomogram into multiple non-overlapping cubes. However, we find that it is difficult for DNNs to learn to pick particles at cube boundaries because their information is split among multiple cubes and therefore is incomplete in each cube. To solve this problem, we set extra overlapping zones for the cubes so that complete particle information is preserved in each cube for training and detection. The width of the overlapping zones is empirically determined to be on the same scale as the average radius of particles. As an example, to process a tomogram of size 512×512×200, we first crop it to a 480×480×192 block. Then the block is divided into 400 cubes with a size of 48× 48×48 each. If we are to give each cube an overlapping zone of width 8, we pad the cube to 64×64×64 using data from its neighboring cubes. Although we use the 64×64×64 cubes for training, only those particles in the center 48×48×48 space, which we refer to as effective detection zones, are used for particle localization and classification so that their complete information is available. Particles outside this area are excluded from calculation of loss function and model prediction.

2.2. Detection Framework

Our detection framework consists of three modules: a DNN particle detection model, a prediction converter, and a loss calculator. The detection model is a 3D adaption of YOLOv3 [23]. Its architecture is shown in Figure 1. It first uses a convolutional block and a series of ResNet blocks [24], referred to as “res” blocks, that function similarly as feature pyramid networks to extract features at multiple scales. To reduce the loss of information in downsampling layers, the model uses convolutional layers with stride 2 instead of max pooling. Extracted deep features are upsampled by nearest neighbor interpolation and then fused with higher-level features. After a series of convolution, the output is used by another higher-level layer. Because input cubes with size 64 ×64×64 are small, the size of particles cannot be too large. To simplify the network architecture, we only use the predictor header at the shallowest level, which is designed to detect small objects. We also reduce the number of downsampling layers to preserve more shallow features. The overall network architecture is similar to that of a U-Net but with fewer decoder layers. The prediction of particles is based on grid cells, not on voxels. For a 64×64×64 input cube, the output of this network consists of 16×16×16 grid cells, with each grid cell representing a 4×4×4 space of the input cube. We choose that one grid cell contains 3 anchor boxes to predict different sizes of particles. These boxes are distinguished by their sizes, which are used as bounding box priors to reduce the difficulty of learning features. The anchor box sizes are calculated by *k*-means clustering on particle sizes of the dataset processed before network training.

To describe a detected particle, each bounding box uses a vector that consists of 3 parts, as shown in Figure 2. The first part contains 4 parameters $t_x, t_y, t_z, t_s$where (t_x, t_y, t_z) denotes the center of the bounding box as the particle location, and t_s denotes the size of the bounding box. To limit the range of the location, t_x, t_y, t_z are sigmoid functions of the detection model output. The second part use one parameter t_{conf} to denote bounding box confidence, which is also the sigmoid function of the detection model output. The last part is a one-hot vector for classification, with each parameter associated with one class. Because each particle belongs to only one class, the vector is the softmax of the detection model output.

The prediction converter is a translation module that converts network output to bounding box coordinates using the following equations:

$$\begin{bmatrix} b_x \\ b_y \\ b_z \end{bmatrix} = \left(\begin{bmatrix} t_x \\ t_y \\ t_z \end{bmatrix} + \begin{bmatrix} c_x \\ c_y \\ c_z \end{bmatrix} \right) * 4 \tag{1}$$

$$b_s = A_s * e^{t_s} \tag{2}$$

Here (c_x, c_y, c_z) denotes offset from the top left of the input cube to the grid cell, and A_s denotes the size of the anchor box in the grid cell. The multiplier 4 corresponds to the size of the grid cell. By using Equations 1 and 2 to convert network output, we finally get the actual predicted particle position (b_x, b_y, b_z) and the real bounding box size b_s associated with the input data. These parameters are used to classify prediction boxes in the loss calculator in the training stage.

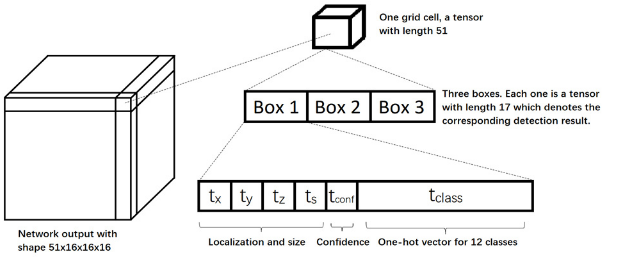
One grid cell, a tensor with length 51
Three boxes. Each one is a tensor with length 17 which denotes the corresponding detection result.
Box 1
Box 2
Box 3
tx
ty
tz
ts
tconf
tclass
Network output with shape 51x16x16x16
Localization and size
Confidence
One-hot vector for 12 classes

Figure 2. The relationship between detection network output and prediction boxes.

In the detection stage, boxes with $t_{conf} > 0.5$ will be picked as candidates. After removing multiple hits by non-maximum suppression (NMS), the candidates from all batches will be mapped according to the associated cube's location to generate the total prediction list of the full tomogram. Another NMS is used to reduce same hits in different cubes. The final list is the output of the entire detection framework.

2.3. Training Strategies and Loss Function

After the prediction converter transforms the network output to bounding boxes, the predictions are divided into three classes for loss calculation according to [23]. In summary, the prediction box that has the largest intersection-over-union (IOU) with any ground truth box is assigned to the class of "positive samples". The remaining prediction boxes that have IOU larger than a selected threshold with any ground truth boxes are assigned to the class of "ignored samples". In this study we set the threshold at 0.333, and the rationale is explained in the Supplementary Material. The prediction boxes left are assigned to the class of "negative samples". Our loss function is the sum of three parts: bounding box loss L_{bbox}, classification loss L_{class}, and confidence loss L_{conf} (Equations 3-6). Our training strategy is that only positive samples contribute to bounding box loss, classification loss and confidence loss. Negative samples only contribute to confidence loss, while ignored samples do not contribute to any loss. By adding ignored samples, the value of confidence in grid cells becomes smoother. Without this design, the object detection model tends to predict all boxes with low confidence. The ignored samples will be removed by non-maximum suppression in the detection stage.

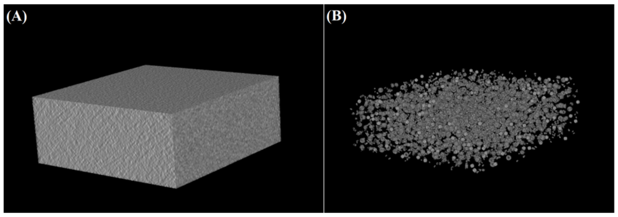

Figure 3. Tomogram 9 from the SHREC20 dataset. (A) The simulated 3D tomogram. (B) Noise-free ground truth for the tomogram in (A).

We use mean square errors (MSE) for calculating bounding box loss and use binary cross entropy loss (BCE) for calculating classification loss and confidence loss. As the number of prediction boxes is much higher than the number of ground truth, to reduce computational cost, we invert Equations 1 and 2 and use the inverted formulas to calculate ground-truth $\hat{t}_x, \hat{t}_y, \hat{t}_z, \hat{t}_s$ from the annotation $\hat{b}_x, \hat{b}_y, \hat{b}_z, \hat{b}_s$ in the training set. This data is used to calculate the bounding box loss in Equation 3. The total loss is defined as follows:

$$L_{bbox} = \lambda_{bbox} \sum_{i=0}^{N} 1_i^{obj} \left[(t_x - \hat{t}_x)^2 + \left(t_y - \hat{t}_y\right)^2 + (t_z - \hat{t}_z)^2 + (t_s - \hat{t}_s)^2\right] \tag{3}$$

$$L_{conf} = -\lambda_{obj} \sum_{i=0}^{N} 1_i^{obj} \ln(t_{conf}) - \lambda_{noobj} \sum_{i=0}^{N} 1_i^{noobj} \ln\left(1 - t_{conf}\right) \tag{4}$$

$$L_{class} = -\lambda_{class} \sum_{i=0}^{N} 1_i^{obj} \sum_{c \in classes} \left[p_i(c) \ln t_{class} + \left(1 - p_i(c)\right) \ln(1 - t_{class})\right] \tag{5}$$

$$L_{total} = L_{bbox} + L_{conf} + L_{class} \tag{6}$$

Here N is the number of predicted bounding boxes and $p_i(c)$ is 1 if the ground truth class is c otherwise 0. We change the weights of different losses by adjusting their corresponding coefficients λ_{bbox}, λ_{obj}, and, λ_{class} , respectively.

3. EXPERIMENTS

3.1. Datasets and Experiments

Our detection framework is evaluated on SHREC2020, a public dataset that consists of ten simulated tomograms of subcellular scale volumes for a competition between participating particle picking algorithms [2]. A sample view of this dataset is shown in Figure 3. Within each tomogram, there are twelve different types of macromolecular complexes whose sizes range from 5 to 19 voxels. Our model has also been evaluated on real cryo-ET images in the EMPIAR-10045 ribosome dataset [25], with detailed results provided in the Supplementary Material. We use precision, recall and F1 score to quantify localization and classification performance of our model. For ground truth, we use data provided by the SHREC2020 competition. To determine whether a prediction is a hit, we follow the same evaluation criteria used in the competition [2]. Specifically, if a predicted location is within the voxels of the dilated ground truth of particles, it is considered as a true positive. To reduce the influence of the background, as most predicted grid cells contain no particles, we set λ_{noobj} to 0.1, λ_{obj} and λ_{conf} to 1, λ_{bbox} and λ_{conf} to 5. The anchor box sizes chosen are 5, 10 and 15 to fit different sizes of particles. We use the Adam optimizer and set the learning rate as 0.001. Our detection framework is implemented using Pytorch 1.7.0 based on Python 3.7.3 and is trained on 4 NVIDIA GeForce RTX 3090. It takes approximately 1 hour and 40 minutes to complete 120 epochs of training.

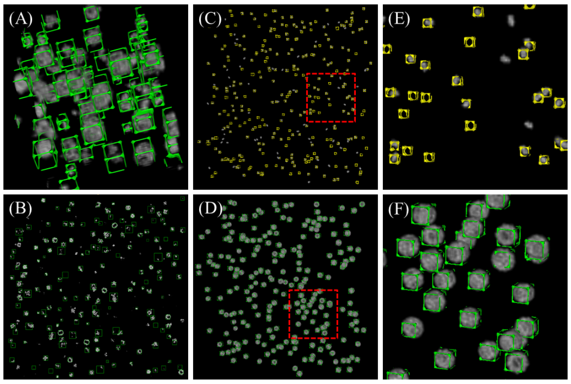

Figure 4. Samples of detection results with location and geometry of actual particles. (A) A cube cropped from the evaluation tomogram. (B) A sample view of one slice. (C) Localization result of molecule 3gl1, an example of small particles. (D) Localization result of molecule 4d8q, an example of large particles. (E) and (F) Magnified views of the red dashed boxes from (C) and (D), respectively.

	Precision	Recall	F1 Score
3D MS-D	0.947	0.906	0.926
DeepFinder	0.957	0.893	0.924
3D ResNet	0.692	0.712	0.702
YOPO	0.901	0.914	0.907
Dn3DUnet	0.900	0.841	0.869
UMC	0.950	0.949	0.949
Ours	0.957	0.947	0.952

Table 1. Results of localization evaluation on SHREC2020 dataset. Results of the other models are reproduced from [2].

	Training stage	Detection stage
3D MS-D	168 h	5 min
DeepFinder	50 h	20 min
3D ResNet	5 h	2 h
YOPO	8 h	40 min
Dn3DUnet	15 h 10 min	1 min 41 s
UMC	16 h	42 min
Ours	1 h 40 min	10 s

Table 2. Time cost of training and test.

	1s3x	3qm1	3gl1	3h84	2cg9	3d2f	1u6g	3cf3	1bxn	1qvr	4cr2	4d8q
3D MS-D	0.192	0.408	0.437	0.416	0.368	0.461	0.492	0.719	0.948	0.851	0.942	0.964
DeepFinder	0.610	0.729	0.800	0.911	0.783	0.848	0.866	0.939	1.000	0.984	0.993	0.993
3D ResNet	0.193	0.185	0.405	0.407	0.334	0.445	0.491	0.628	0.906	0.719	0.868	0.817
YOPO	0.558	0.741	0.670	0.834	0.696	0.682	0.795	0.896	0.987	0.83	0.923	0.993
Dn3DUnet	0.529	0.577	0.569	0.674	0.332	0.523	0.462	0.676	0.925	0.684	0.907	0.974
UMC	0.661	0.827	0.839	0.947	0.855	0.873	0.899	0.981	0.997	0.980	1.000	0.997
Ours	0.397	0.610	0.675	0.599	0.552	0.634	0.695	0.845	0.971	0.875	0.964	0.985

Table 3. Comparison of F1 scores of our model versus competing methods. The first row lists the PDB ID of each macromolecule.

3.2. Results

Performance of our model in comparison with other models submitted to the SHREC2020 competition is summarized in Table 1. The precision of our model is higher than that of the competing models, while the recall of our model ranks the second. Comparison of classification performance is summarized in Table 3. Representative results are shown in Figure 4. Because our model uses only the coordinates of particles instead of their geometry, it is less accurate in detecting and distinguishing aspherical particles. Experiments shows that the recall rates for large globular particles such as 4d8q, 1xbn, 3cf3 are good but poor for small aspherical particles such as 1s3x and 3qm1. Comparison of training and testing cost in terms of processing time is summarized in Table 2. Our model is approximately ten times faster than the fastest segmentation model. Influence of overlap size on performance is examined in the Supplementary Material. With sufficient overlap size, the model can predict more particles and achieves higher recall and IOU. However, when the overlap size is too large, the number of particles whose centers are in overlap zones increases. The detection performance will be negatively impacted because these particles are considered as background by our training strategy.

On EMPIAR-10045, a dataset of real cryo-ET images, our model achieves a precision of 0.69 and a recall of 0.75 in localization, whereas a 3D U-Net with connected component analysis provides a precision of 0.68 and a recall of 0.87. Further details can be found in the Supplementary Material.

4. CONCLUSION

In this study, we have developed a one-stage object detection model for fast particle picking from 3D tomograms. Compared with the state-of-the-art segmentation models, our model is substantially faster because it combines bounding box regression and classification in one step, which is key to its fast speed. Our model provides an efficient tool for processing large 3D tomograms. However, our study also has its limitations. The poor classification accuracy of our model for small particles indicates that more shallow features should be preserved. Another limitation is that the size of the grid cell limits the minimum distance between two particles, making it difficult for the model to distinguish between adjoining particles. These limitations will be addressed in our future work.

5. ACKNOWLEDGMENTS

This study was supported in part by the Strategic Priority Research Program of the Chinese Academy of Sciences (grant XDB37040402) and the National Natural Science Foundation of China (grant 31971289, 91954201).

6. COMPLIANCE WITH ETHICAL STANDARDS

This is a numerical simulation study for which no ethical approval was required.

7. REFERENCES

[1] M. Turk and W. Baumeister, "The promise and the challenges of cryo-electron tomography," *FEBS letters,* vol. 594, no. 20, pp. 3243-3261, 2020.

[2] I. Gubins *et al.*, "SHREC 2020: Classification in cryo-electron tomograms," *Computers & Graphics,* vol. 91, pp. 279-289, 2020.

[3] W. V. Nicholson and R. M. Glaeser, "Review: Automatic particle detection in electron microscopy," *Journal of Structural Biology,* vol. 133, no. 2, pp. 90-101, 2001.

[4] Z. Huang and P. A. Penczek, "Application of template matching technique to particle detection in electron micrographs," *Journal of Structural Biology,* vol. 145, no. 1, pp. 29-40, 2004.

[5] A. S. Frangakis *et al.*, "Identification of macromolecular complexes in cryoelectron tomograms of phantom cells," *Proceedings of the National Academy of Sciences,* vol. 99, no. 22, pp. 14153-14158, 2002.

[6] J. Böhm, A. S. Frangakis, R. Hegerl, S. Nickell, D. Typke, and W. Baumeister, "Toward detecting and identifying macromolecules in a cellular context: Template matching applied to electron tomograms," *Proceedings of the National Academy of Sciences,* vol. 97, no. 26, pp. 14245-14250, 2000.

[7] R. J. Hall and A. Patwardhan, "A two step approach for semi-automated particle selection from low contrast cryo-electron micrographs," *Journal of Structural Biology,* vol. 145, no. 1, pp. 19-28, 2004.

[8] N. R. Voss, C. K. Yoshioka, M. Radermacher, C. S. Potter, and B. Carragher, "DoG Picker and TiltPicker: Software tools to facilitate particle selection in single particle electron microscopy," *Journal of Structural Biology,* vol. 166, no. 2, pp. 205-213, 2009.

[9] D. Woolford, B. Hankamer, and G. Ericksson, "The Laplacian of Gaussian and arbitrary z-crossings approach applied to automated single particle reconstruction," *Journal of Structural Biology,* vol. 159, no. 1, pp. 122-134, 2007.

[10] Z. Zhao, P. Zheng, S. Xu, and X. Wu, "Object detection with deep learning: A Review," *IEEE Transactions on Neural Networks and Learning Systems,* vol. 30, no. 11, pp. 3212-3232, 2019.

[11] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, "You only look once: Unified, real-time object detection," in *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 779-788, 2016.

[12] W. Liu *et al.*, "SSD: Single shot multi-box detector," in *2016 European Conference on Computer Vision (ECCV)*, pp. 21-37, 2016.

[13] R. Girshick, "Fast R-CNN," in *2015 IEEE International Conference on Computer Vision (ICCV)*, pp. 1440-1448, 2015.

[14] S. Ren, K. He, R. Girshick, and J. Sun, "Faster R-CNN: towards real-time object detection with region proposal networks," *IEEE Transactions on Pattern Analysis and Machine Intelligence,* vol. 39, no. 6, pp. 1137-1149, 2017.

[15] K. He, G. Gkioxari, P. Dollár, and R. Girshick, "Mask R-CNN," in *2017 IEEE International Conference on Computer Vision (ICCV)*, pp. 2980-2988, 2017.

[16] F. Wang *et al.*, "DeepPicker: A deep learning approach for fully automated particle picking in cryo-EM," *Journal of Structural Biology,* vol. 195, no. 3, pp. 325-336, 2016.

[17] T. Wagner *et al.*, "SPHIRE-crYOLO is a fast and accurate fully automated particle picker for cryo-EM," *Communications Biology,* vol. 2, no. 1, p. 218, 2019, doi: 10.1038/s42003-019-0437-z.

[18] R. Li *et al.*, "Automatic localization and identification of mitochondria in cellular electron cryo-tomography using faster-RCNN," *BMC Bioinformatics,* vol. 20, no. 3, p. 132, 2019, doi: 10.1186/s12859-019-2650-7.

[19] Ö. Çiçek, A. Abdulkadir, S. S. Lienkamp, T. Brox, and O. Ronneberger, "3D U-Net: Learning dense volumetric segmentation from sparse annotation," in *2016 Medical Image Computing and Computer-Assisted Intervention (MICCAI)*, pp. 424-432, 2016.

[20] O. Ronneberger, P. Fischer, and T. Brox, "U-Net: Convolutional networks for biomedical image segmentation," in *2015 Medical Image Computing and Computer-Assisted Intervention (MICCAI)*, N. Navab, J. Hornegger, W. M. Wells, and A. F. Frangi, Eds., pp. 234-241, 2015.

[21] E. Moebel *et al.*, "Deep learning improves macromolecule identification in 3D cellular cryo-electron tomograms," *bioRxiv,* 2020, doi: 10.1101/2020.04.15.042747.

[22] Y. Hao *et al.*, "VP-Detector: A 3D convolutional neural network for automated macromolecule localization and classification in cryo-electron tomograms," *bioRxiv,* 2021, doi: 10.1101/2021.05.25.443703.

[23] A. Farhadi and J. Redmon, "Yolov3: An incremental improvement," in *Computer Vision and Pattern Recognition*, 2018: Springer Berlin/Heidelberg, Germany, p. 1804.02767.

[24] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 770-778, 2016.

[25] T. A. M. Bharat and S. H. W. Scheres, "Resolving macromolecular structures from electron cryo-tomography data using subtomogram averaging in RELION," *Nature Protocols,* vol. 11, no. 11, pp. 2054-2065, 2016, doi: 10.1038/nprot.2016.124.