Convolutional neural networks for automated annotation of cellular cryo-electron tomograms

Muyuan Chen[1,2], Wei Dai[2,4], Stella Y Sun[2], Darius Jonasch[2], Cynthia Y He[3], Michael F Schmid[2], Wah Chiu[2] & Steven J Ludtke[2]

Cellular electron cryotomography offers researchers the ability to observe macromolecules frozen in action *in situ*, but a primary challenge with this technique is identifying molecular components within the crowded cellular environment. We introduce a method that uses neural networks to dramatically reduce the time and human effort required for subcellular annotation and feature extraction. Subsequent subtomogram classification and averaging yield *in situ* structures of molecular components of interest. The method is available in the EMAN2.2 software package.

Cellular electron cryotomography (CryoET) is the dominant technique for studying the structure of interacting, dynamic complexes in their native cellular environment at nanometer resolution. While fluorescence imaging techniques can provide high-resolution localization of labeled complexes within the cell, they cannot determine the structure of the molecules themselves. X-ray crystallography and single-particle CryoEM allow researchers to study the high-resolution structures of macromolecules, but these molecules must first be purified, and they cannot be studied *in situ*. In the cellular environment, complexes are typically transient and dynamic; thus CryoET provides information about cellular processes that is not attainable by any other current method.

The major limiting factors in cellular CryoET data interpretation include high noise levels, the missing wedge artifact due to experimental geometry (**Supplementary Fig. 1**), and the challenge of studying crowded macromolecules that undergo continuous conformational changes[1]. A great deal can be learned by simply annotating the contents of the cell and observing spatial inter-relationships among macromolecules. Beyond this, subvolumes can be extracted and aligned to improve resolution by reducing noise and eliminating missing wedge artifacts[2,3]. Unfortunately, such averages are often limited by the extensive conformational and compositional variability within the cell[4,5], and thus classification is required to achieve more homogeneous populations. Before particles can be extracted and averaged, (macro)molecules of one type must be identified with high fidelity. This task, and the broader task of annotating the contents of the cell, is typically performed by human annotators; it is extremely labor intensive—as much as one week is required for an expert to annotate a typical (4,000 × 4,000 × 1,000 voxel) tomogram. With automated methods now able to produce many cellular tomograms per microscope per day, annotation has become a primary time-limiting factor in the processing pipeline[6].

While algorithms have been developed for automatic segmentation of specific features (e.g., refs. 7–9), each class of feature has typically required a separate development effort, and a generalizable algorithm for arbitrary feature recognition has been lacking. We have developed a method based on convolutional neural networks (CNN) that is capable of learning a wide range of possible feature types and effectively replicates the behavior of a specific expert human annotator. The network requires only minimal human training, and the structure of the network itself is fixed. This method can readily discriminate subtle differences such as double membranes of mitochondria versus single membranes of other organelles. This single algorithm works well on the major classes of geometrical objects: extended filaments such as tubulin or actin, membranes (curved/planar surfaces), periodic arrays and isolated macromolecules.

Deep neural networks have been broadly applied across many applications in recent years[10]. Past CryoEM applications of neural networks have been limited to simpler methods developed before deep learning, which do not perform well on tomographic data. Among the various deep neural network concepts, deep CNNs are especially useful for pattern recognition in images. While ideally we would develop a single network capable of annotating all known cellular features, the varying noise levels, different artifacts and features in different cell types, and large computational requirements make this impractical at present.

A more tractable approach is to simplify the problem by training one CNN for each feature to be recognized, then merging the results from multiple networks into a single multicomponent annotation (**Fig. 1** and **Supplementary Video 1**). We have successfully designed a CNN with only a few layers that contains wider than typical kernels (**Supplementary Fig. 2**) which can, with minimal training, successfully identify a wide range of different features across a diverse range of cellular tomograms. The network can be trained quickly, with as few as 10 manually annotated image tiles (64 × 64 pixels) containing the feature of interest and 100 tiles lacking the targeted feature. The CNN-based method we have

[1]Graduate Program in Structural and Computational Biology and Molecular Biophysics, Baylor College of Medicine, Houston, Texas, USA. [2]Verna Marrs and McLean Department of Biochemistry and Molecular Biology, Baylor College of Medicine, Houston, Texas, USA. [3]Department of Biological Science, Centre for BioImaging Sciences, National University of Singapore, Singapore. [4]Present address: Department of Cell Biology and Neuroscience, Center for Integrative Proteomics Research, Rutgers University, Piscataway, New Jersey, USA. Correspondence should be addressed to S.J.L. (sludtke@bcm.edu).

RECEIVED 20 JANUARY; ACCEPTED 13 JULY; PUBLISHED ONLINE 28 AUGUST 2017 ; DOI:10.1038/NMETH.4405

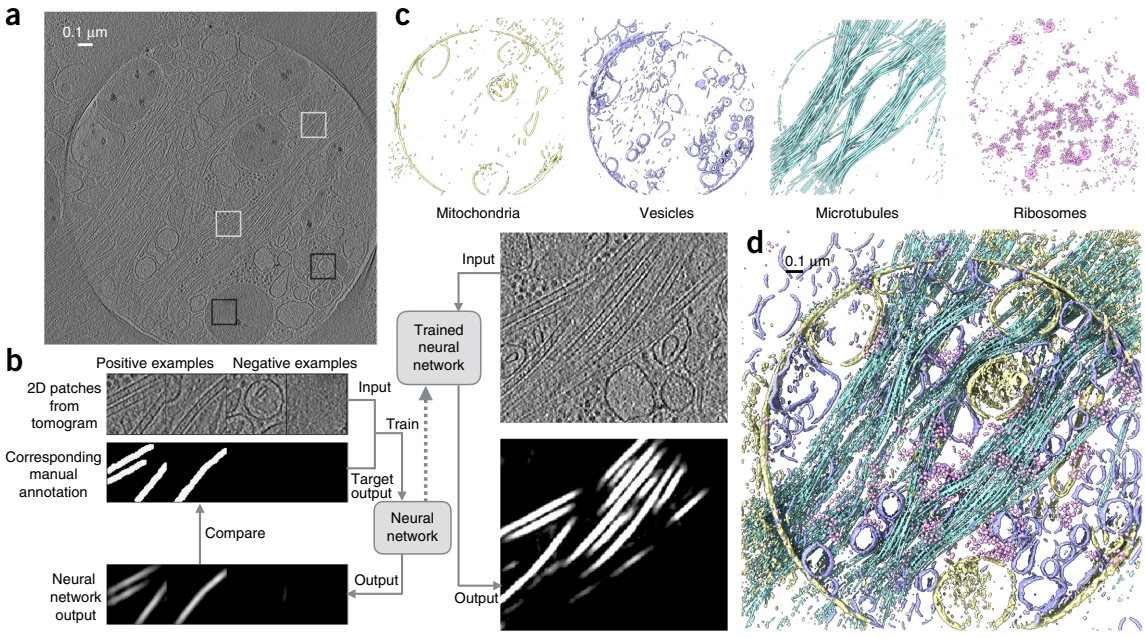

Figure 1 | Workflow of automated tomogram annotation using a PC12 cell as an example. (**a**) Slice view of a raw tomogram input. Locations of representative positive (white) and negative (black) examples are shown as boxes. (**b**) Training and annotation of a single feature. Representative 2D patches of both positive (10 total) and negative examples (86 total) are extracted and manually segmented. The time required for manual selection and annotation was ~5 min. Tomogram patches and their corresponding annotations are used to train the neural network, so that the network outputs match manual annotations. The trained neural network is applied to whole 2D slices, and the feature of interest is segmented. (**c**) Annotation of multiple features in a tomogram. Four neural networks are trained independently to recognize double membrane (yellow), single membrane (blue), microtubule (cyan) and ribosome (pink). (**d**) Masked-out density of the merged final annotation of the four features.

developed operates on the tomogram slice by slice, rather than in 3D, similar to most current manual annotation programs. This approach greatly reduces the complexity of the neural network, improves computational speed and largely avoids the distortions due to the missing wedge artifact[1] (**Supplementary Fig. 1**).

Once a CNN has been trained to recognize a certain feature, it can be used to annotate the same feature in other tomograms of the same type of cell under similar imaging conditions without additional training. We have found that training on one tomogram is generally adequate to successfully annotate all of the tomograms within a particular biological project (see **Supplementary Fig. 3**). This enables rapid annotation of large numbers of cells of the same type with minimal human effort and a reasonable amount of computation.

When multiple scientists are presented with the same tomogram, annotation results are not identical[11,12]. While results are grossly similar in most cases, the annotation of specific voxels can vary substantially among users. Nonetheless, when presented with each other's results, most annotators agree that alternative annotations were also reasonable. That is to say, it is extremely difficult to establish a single 'ground truth' in any cellular annotation process. Neural networks make it practical to explore this variability. Rather than having multiple annotators train a single CNN, a CNN can be trained for each annotator for each feature of interest to produce both a consensus annotation as well as a measure of uncertainty. Given the massive time requirements for manual annotation, this kind of study would never be practical on a large scale using real human annotations; but to the extent that a trained CNN can mimic a specific annotator, this sort of virtual comparison is a practical alternative.

To test our methodology, we used four distinct cell types: PC12 cells, human platelets[13], African trypanosomes and cyanobacteria[14]. We targeted multiple subcellular features with various geometries, including bacteriophages, carboxysomes, microtubules, double-layer membranes, full and empty vesicles, RuBisCO molecules, and ribosomes. Representative annotations are shown in **Figure 2a–c**, **Supplementary Fig. 4** and **Supplementary Videos 2–5**. These data sets were collected on different microscopes with different defocus and magnification ranges and with or without a phase plate (**Supplementary Table 1**).

After annotation, subtomograms of specific features of interest were extracted from each tomogram, and subtomogram averaging was applied to each set in order to test the usefulness of the annotation. Results are shown in **Figure 2d–f**. For instance, the subtomogram average of automatically extracted ribosomes from trypanosomes resembles an ~40-Å-resolution version of that structure determined by single-particle CryoEM[15] (**Fig. 2f**). The structures of microtubules have the familiar features and spacing observed in low-resolution structures[16,17], while some have luminal density, and others do not[18] (**Fig. 2e**). We are able to detect and trace the en face aspect of the thylakoid membrane in cyanobacteria, which is known to have a characteristic pseudocrystalline array[19] of light-harvesting complexes embedded in it, in both the subtomogram patches and their corresponding power spectra (**Fig. 2d**). These results confirm that the methodology is correctly identifying the subcellular features being trained with a high level of accuracy. Our example tomograms were collected at low magnification for annotation and qualitative cellular biology, not with subtomogram averaging in mind, and thus the resolution of the averages is limited.

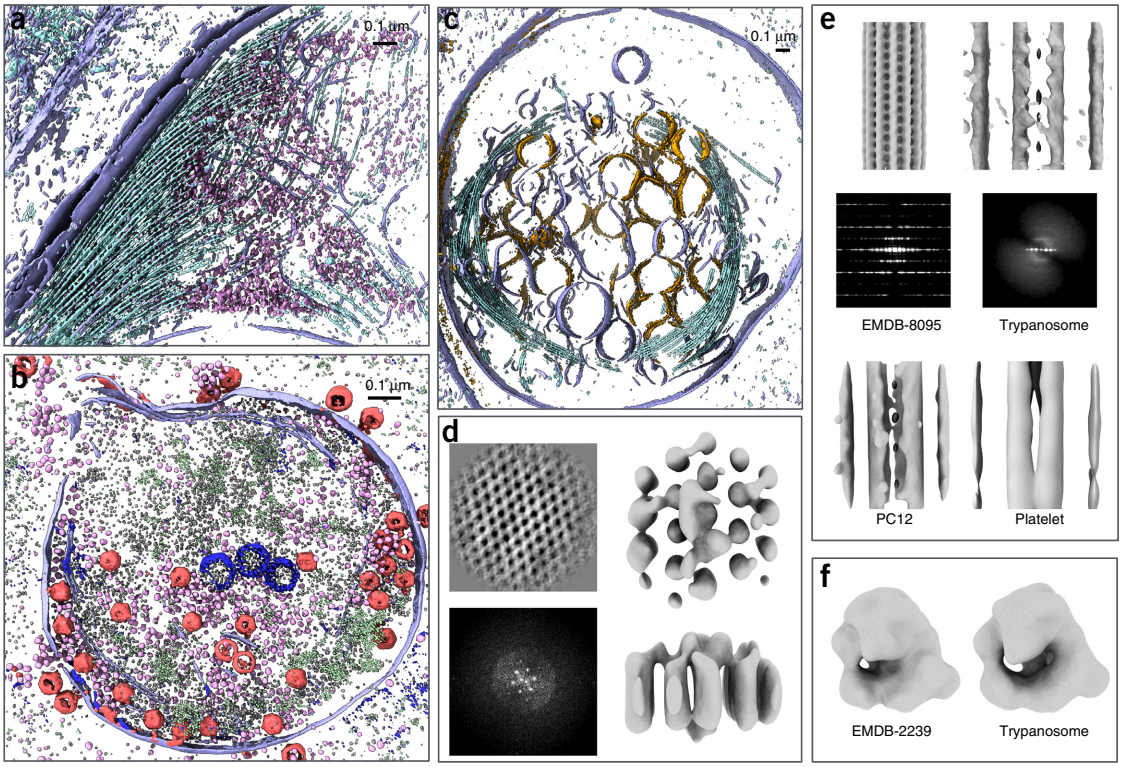

Figure 2 | Results from automated tomogram annotation. (**a**–**c**) Masked-out density of tomogram annotation results. (**a**)Trypanosome. Purple, membrane; cyan, microtubules; pink, ribosomes. (**b**) Cyanobacteria. Purple, vertical membranes; pink, ribosomes; red, phages; white, RubisCOs; dark blue, carboxysomes; green, proteins on thylakoid membrane. (**c**) Platelet. Purple, membranes with similar intensity value on both sides orange, membranes of vesicles with darker density inside; cyan, microtubules. (**d**–**f**) Subtomogram averaging of extracted particles. (**d**) Horizontal thylakoid membrane in cyanobacteria. Top left, 2D class average of particle projections; bottom left, Fourier transform (FFT) of 2D class average; top right, top view of 3D average; bottom right, side view of 3D average. (**e**) Microtubules. Top-left, *in vitro* CryoEM microtubule structure (EMDB 8095), low-pass filtered to 40 Å; top right, average of trypanosome microtubules from CryoET data (N = 511). Density on the left and right resembles adjacent microtubules; middle left, FFT of projection of the EMDB structure; middle right, incoherent average of FFT of projection of trypanosome microtubules; bottom left, average of PC12 microtubules (N = 677); bottom right, average of platelet microtubules (N = 312). (**f**) Ribosomes. Left, *in vitro* CryoEM ribosome structure (EMDB 2239), low-pass filtered to 40 Å; right, average of trypanosome ribosomes from CryoET data (N = 759).

The set of annotation utilities is freely available as part of the open-source package EMAN2.2 (ref. 20) (http://www.EMAN2.org), and it includes a graphical interface for training new CNNs as well as applying them to tomograms.

METHODS

Methods, including statements of data availability and any associated accession codes and references, are available in the online version of the paper.

Note: Any Supplementary Information and Source Data files are available in the online version of the paper.

ACKNOWLEDGMENTS

We gratefully acknowledge support of NIH grants (R01GM080139, P01NS092525, P41GM103832), Ovarian Cancer Research Fund and Singapore Ministry of Education. Molecular graphics and analyses performed with UCSF ChimeraX, developed by the Resource for Biocomputing, Visualization, and Informatics at the University of California, San Francisco.

AUTHOR CONTRIBUTIONS

M.C. designed the protocol. W.D., S.Y.S. and C.Y.H. provided the test data sets. M.C. and D.J. tested and refined the protocol. M.C., W.D., S.Y.S., M.F.S., W.C. and S.J.L. wrote the paper and provided suggestions during development.

COMPETING FINANCIAL INTERESTS

The authors declare no competing financial interests.

1. Lučič, V., Rigort, A. & Baumeister, W. *J. Cell Biol.* **202**, 407–419 (2013).
2. Galaz-Montoya, J.G. *et al. J. Struct. Biol.* **194**, 383–394 (2016).
3. Chen, Y., Pfeffer, S., Hrabe, T., Schuller, J.M. & Förster, F. *J. Struct. Biol.* **182**, 235–245 (2013).
4. Asano, S. *et al. Science* **347**, 439–442 (2015).
5. Pfeffer, S., Woellhaf, M.W., Herrmann, J.M. & Förster, F. *Nat. Commun.* **6**, 6019 (2015).
6. Ding, H.J., Oikonomou, C.M. & Jensen, G.J. *J. Struct. Biol.* **192**, 279–286 (2015).
7. Rigort, A. *et al. J. Struct. Biol.* **177**, 135–144 (2012).
8. Page, C., Hanein, D. & Volkmann, N. *Ultramicroscopy* **155**, 20–26 (2015).
9. Frangakis, A.S. *et al. Proc. Natl. Acad. Sci. USA* **99**, 14153–14158 (2002).
10. LeCun, Y., Bengio, Y. & Hinton, G. *Nature* **521**, 436–444 (2015).
11. Garduño, E., Wong-Barnum, M., Volkmann, N. & Ellisman, M.H. *J. Struct. Biol.* **162**, 368–379 (2008).
12. Hecksel, C.W. *et al. Microsc. Microanal.* **22**, 487–496 (2016).
13. Wang, R. *et al. Proc. Natl. Acad. Sci. USA* **112**, 14266–14271 (2015).
14. Dai, W. *et al. Nature* **502**, 707–710 (2013).
15. Hashem, Y. *et al. Nature* **494**, 385–389 (2013).
16. Asenjo, A.B. *et al. Cell Rep.* **3**, 759–768 (2013).
17. Koning, R.I. *et al. J. Struct. Biol.* **161**, 459–468 (2008).
18. Garvalov, B.K. *et al. J. Cell Biol.* **174**, 759–765 (2006).
19. Scheuring, S. & Sturgis, J.N. *Science* **309**, 484–487 (2005).
20. Tang, G. *et al. J. Struct. Biol.* **157**, 38–46 (2007).

ONLINE METHODS

Preprocessing. Raw tomograms tend to be extremely noisy, and while the CNN can perform filter-like operations, long range-filters are more efficiently performed in Fourier space, so this is handled as a preprocessing step. A low-pass filter (typically at 20 Å) and high-pass filter (typically at 4,000 Å) are performed—the first to reduce noise and the second to reduce ice gradients, which may interfere with feature identification. The tomogram is also normalized such that the mean voxel value is 0.0, and the s.d. is 1.0. Voxels with intensity higher or lower than three times the s.d. are clamped. These preprocessing steps provide a consistent input range for the CNN.

Training sets. The CNN is trained using 64 × 64 pixel tiles manually extracted from the tomogram using a specialized graphical tool in EMAN2. Roughly ten tiles containing the feature of interest must be selected and then manually annotated with a binary mask using a simple drawing tool. Since this is the only information used to train the CNN, it is important to span the range of observed variations for any given feature when selecting tiles to manually annotate. For example, when training for membranes, segmentation performance will be improved if a range of curvatures are included in training. An additional roughly 100 tiles not containing the feature of interest must also be selected as negative examples. Since no annotation is required for these regions, they can be selected very rapidly. The larger number of negative examples is required to cover the diversity of features in the tomogram which are not the feature of interest. The size of 64 × 64 pixels is fixed in the current design as a practical trade-off between accuracy, computational efficiency and GPU memory requirements.

Image rotation is not explicitly part of the network structure, so each of the positive training patches is automatically replicated in several random orientations to reduce the number of required manual annotations as well as to compensate for anisotropic artifacts in the image plane. Examples containing the feature of interest are replicated such that the total number of positive and negative examples is roughly equal for training.

Rationale of neural network design. Consider how per-pixel feature recognition would be handled with a traditional neural network. Clearly a pixel cannot be classified strictly based on its own value. To identify the feature it represents, we must also consider some number of surrounding pixels. For example, if we considered a 30 × 30 pixel region with the pixel under consideration in the center, that would put 900 neurons in the first layer of the neural network, and full connectivity to a second layer would require 900^2 weights. The number of neurons could be reduced in later layers. However, to give each pixel the advantage of the same number of surrounding pixels, we would have to run a 30 × 30 pixel tile through the network for each input pixel in the tomogram. For a 4,000 × 4,000 × 1,000 tomogram, assuming a total of ~5×10^6 weights in the network and a simple activation function, this would require 10^{17}–10^{18} floating point operations. That is, annotation of a single tomogram would require days to weeks, even with GPU technology.

Similar to the use of FFTs to accelerate image processing operations, the concept of CNNs allows process volumes to be handled much more efficiently. In a CNN the concept of a neuron is extended such that a single neuron takes an image as its input, and a connection between neurons becomes a convolution operation. These more complicated neurons give the overall network organization a much simpler appearance, since some complexity is hidden within the neurons themselves.

Rotational invariance in the network is handled, as described above, by providing representative tiles in many different orientations and is one of the major reasons we require 40 neurons. However, the network also needs to locate features irrespective of their translational position within the tile. The use of CNNs in our network design effectively performs a translation-independent classification for every pixel in the image simultaneously. That is, it is possible to process an entire slice of a tomogram in a single operation, rather than a tile for every pixel to be classified in the traditional approach.

The concept underlying deep neural networks is that, by combining many layers of very simple perceptron units, it is possible to learn arbitrarily complex features in the data. In addition to dramatically simplifying the network training process, the use of many layers permits arbitrary nonlinear functions to be represented despite the use of a simple ReLU activation function[21]. Many new techniques have been developed around this concept[22–24], and these techniques improve on the convergence rate and robustness of classical neural network designs, such as the CNN, and so enable training of large networks to solve problems which were previously intractable for neural network technology.

In typical use cases, pretrained neural networks are provided to users, and the users simply apply the existing network to their data[25]. The training process is generally extremely complicated and requires both a skilled programmer and significant computational resources. However, in the current application, pretraining of the network is not feasible on account of the lack of appropriate examples spanning the diverse potential user data and features of interest.

Another challenge of biological feature recognition is that biological features cover a wide range of scales. The pooling technique used in convolutional neural network design is capable of spanning such scale differences, but spatial localization precision is sacrificed during this process. Although it is possible to perform unpooling and train interpolation kernels at the end of the network[26], or simply interpolate by applying the neural network on multiple shifted images, these methods make the training process much slower and make it more difficult to achieve convergence. Given our inability to provide a pretrained network, we needed a design that could be trained reliably in an automatic fashion so end users with no knowledge of neural networks could make use of the system. Instead of a single large and deep neural network that performs the entire classification process, we use a set of smaller independent networks, each of which recognizes only a single feature. These smaller networks are then merged to produce the overall classification result. In addition, to reduce the problems caused by unpooling and permit a shallower structure, we use 15 × 15 pixel kernels, which are larger than those typically used in CNN image processing[27]. This larger kernel size allows the discrimination of fine details such as single-layer versus double-layer membranes, while still taking a large (30 × 30 pixel) neighborhood into account. The number of neurons was selected through extensive testing with real data to be as small as possible while remaining accurate.

doi:10.1038/nmeth.4405

While we would ideally like the CNN to make use of true 3D information and achieve true 3D annotation, there are significant technical difficulties in building a 3D CNN. First, the memory requirements of a 3D CNN are dramatically higher than a 2D CNN with the same kernel and tile sizes. Current generation GPUs, unfortunately, still have substantially less RAM than CPUs, meaning this implementation would require a significantly different structure to be practical. Second, even if available RAM were sufficient, convergence of the CNN would be much harder to achieve on account of the dramatic increase in the number of parameters. Third, rotational invariance is required in the trained network. In 2D this requires accommodating only one degree of freedom, but in 3D this requires covering three, which requires a dramatic increase in the number of neurons in the network. This is to say that the current approach cannot simply be adapted to 3D, and a fundamentally different approach would be required to make such a network practical. However, given the impact of the missing wedge, it is likely that a simpler approach of simply including a small number of slices above and below the annotated slice could achieve similar benefits to those of a true 3D approach and with a tolerable increase in computational costs.

Neural network structure. The four-layer CNN we settled on was used in all of our examples (**Supplementary Fig. 2**). The first layer is a convolutional layer containing 40 neurons, each of which has a 2D kernel with 15 × 15 pixels and a scalar offset value. In the input to each first-layer neuron, particles are filtered by a 2D kernel. A linear offset and a ReLU activation function are applied to the neuron output. The activation function is what permits nonlinear behavior in the network, though the degree of nonlinearity is limited by the small number of layers we are using. Each first-layer neuron thus outputs a 64 × 64 tile. The second layer performs max pooling, which outputs the maximum value in each 2 × 2 pixel square in the input tiles. This downsamples each tile to 32 × 32. The second layer is then fully connected to the 40-neuron third layer, again with independent 15 × 15 convolution kernels for each of the 1,600 connections. Although the size of the kernel is the same as in the first layer, thanks to the max pooling, the kernel in this layer can cover features of a larger scale, up to 30 × 30 pixels, roughly half of the training tile size. The fourth and final layer consists of only one neuron producing the final single 32 × 32 output tile. Finally, the filtered results are summed, and a linear offset is applied. To match the results of a manual annotation and expedite convergence, a specialized ReLU activation function ($y = \min(1,x)$) is used.

Hyperparameter selection and neural network training. Before training, all the kernels in the neural network are initialized using a uniform distribution of near-zero values, and the offsets are initialized to zero. Log-squared residual ($\log((y - y')^2)$) between the neural network output and the manual annotation is used as the loss function. Since there is a pooling layer in the network, the manual annotation is shrunk by two to match the network output. An L1 weight decay of 10^{-5} is used for regularization of the training process. No significant overfitting is observed, likely because the high noise level in the CryoET images also serves as a strong regularization factor. To optimize the kernels, we use stochastic gradient descent with a batch size of 20. By default, the neural network is trained for 20 iterations. The learning rate is set to 0.01 in the first iteration and decreased by 10% after each iteration. The training process can be performed on either a GPU or in parallel on multiple CPUs (~10× slower on our testing machine). Training each feature typically takes under 10 min on a current-generation GPU, and the resulting network can be used for any tomogram of the same cell type collected under similar conditions. A workstation with 96 GB of RAM, 2× Intel X5675 processors for a total of 12 compute cores, and an Nvidia GTX 1080 GPU were used for all testing.

Applying trained convolutional neural networks to tomograms. Since the neural network is convolutional, we simply filter the full-sized prefiltered tomogram slices with the trained kernels in the correct order to generate the output. Unlike the training process, the CNN is applied to entire (typically 4,000 × 4,000) tomogram slices. The network is applied to the images by propagation of the image as described above in the subsection "Neural network structure." Practical implementation involves simple matrix operations combined with FFTs for the convolution operations. The final output tile is unbinned by two to match the dimensions of the input tile. Each voxel in the assembled density map is related to the likelihood that that voxel corresponds to the feature used to train the network. While the networks are trained on the GPU, because of memory limitations and the large number of kernels, applying the trained network is currently done on CPUs, where the code can be efficiently parallelized, assuming sufficient RAM is available. For reference, annotation of one feature on a 4,096 × 4,096 × 1,024 tomogram requires ~20 h on the test computer. Note that the full 4,000 × 4,000 tomogram is only needed when annotating small features that are only visible in the unbinned tomogram. Practically, all features in the test data sets shown in this paper were annotated on tomogram binned by 2–8. As the speed of CNN application scales linearly with the number of voxels in the tomogram, it only takes ~2.5 h to annotate the same tomogram binned by two. However, there is still room for significant optimization of the code through reorganization of mathematical operations without altering the network structure.

Postprocessing and merging features. After applying a single CNN to a tomogram, the output needs to be normalized, so the results from the set of trained networks are comparable. This is done by scaling the output of the neural network annotation so that the mean value on manually annotated regions in positive pixels in the training set is one. After normalization, annotation results from multiple CNNs can be merged by simply identifying which CNN had the highest value for each voxel. We also use 1 as a threshold value for isosurface display as well as for particle identification/extraction for subtomogram averaging.

Particle extraction and averaging. While annotation alone is sufficient for certain types of cellular tomogram interpretation, subtomogram extraction and averaging remain a primary purpose for detailed annotation. To extract discrete objects like ribosomes or other macromolecules, we begin by identifying all connected voxels annotated as being the same feature. For each connected region, the maxima position in the annotation is used as the particle location. For continuous features like microtubules,

we randomly seed points on the annotation output and use these points as box coordinates for particle extraction. In both cases, EMAN2 2D classification[20] is performed on a *z*-axis projection of the particles in order to help identify and remove bad particles in a fashion similar to single-particle analysis. 3D alignment and averaging are performed using EMAN2 single-particle tomography utilities[2]. For the ribosomes, we ran an iterative refinement using a high-resolution ribosome structure (EMDB-2239) phase randomized to 66 Å as an initial model. For the subtomogram averaging of microtubules and thylakoid membranes, we generated initial models from good 2D class averages of a *z*-axis projection of the particles.

Reusability. Once a neural network is trained to recognize a feature in one tomogram, within some limits it can be used to annotate the same feature in other tomograms. These tomograms should have the same voxel size and should have been preprocessed in the same way. For some universal features like the membranes, it would be possible to apply a trained neural network on a completely different cell type, but for most other features, it is strongly recommended that the neural network be trained on a tomogram of the same cell type under similar imaging conditions. In practice, we have found performance of the neural network to be robust to reasonable differences in defocus or signal-to-noise ratio. These statements are difficult to quantify, however, since different types of features have different sensitivity to such factors.

To perform a rough quantification of network reusability, we segmented ribosomes from a single cyanobacteria tomogram collected using a Zernike phase plate[28] using four different CNNs, each trained in a different way (**Supplementary Fig. 3**). We used each CNN to identify ribosomes, and then we manually identified which of the putative ribosomes appeared to be something else. We tested only for false positives, not false negatives. While this is not a robust test, since we lack ground truth, we believe it does at least give a general idea about reusability of CNNs. All four CNNs identified ~800–900 ribosomes in the test tomogram. It should be noted that this test used a single CNN for classification, whereas in a normal use case, multiple CNNs would be competing for each voxel, which would improve classification accuracy. Given this, the achieved accuracy levels are actually quite high. In the first test, we trained a CNN using the test tomogram itself with only 5 positive samples and 50 negative samples. We estimate that 89% of the particles were correctly identified. In the second test, a tomogram from the same set also using the Zernike phase plate was used for training, and the network achieved an estimated 86% accuracy. The third CNN was trained using a tomogram collected with conventional imaging (no phase plate), with an estimated accuracy of 78%. We note that in this test, most of the false positives were due to cut-on artifacts in the phase plate tomogram. The final neural network was again trained on the test tomogram, but this time with 10 positive examples and 100 negative examples. This improved performance to 91%, which somewhat better than that of the first test with fewer training tiles. While ribosomes are somewhat larger and denser than most other macromolecules in the cell, even something somewhat smaller, like a free RuBisCO, could potentially be misidentified if a competitive network were not being applied for this feature. For this reason, we consider ≥75% performance with a single network to be extremely good. We also show a cross-test on microtubules in **Supplementary Figure 3**.

Limitations. As a machine-learning model, the only knowledge input to the CNN is from the provided training set. Although the network is robust to minor variations of the target feature and manual annotation in the training set (**Supplementary Fig. 5**), its performance is unpredictable on features which have not been specifically trained for in any of the CNNs. Some nonbiological features, such as carbon edges and gold fiducials, have much higher contrast than the biological material and may cause locally unstable behavior of the neural network (**Fig. 1c–d**). While misidentification of these features is not generally a problem in display and subtomogram averaging (they can easily be marked as outliers and get removed in the average), it is possible to train a CNN to only recognize these nonbiological features and to effectively remove them from the annotation output (**Supplementary Fig. 6**). In addition, due to the design of the neural network, the maximum scale range of detectable feature is 30 pixels. That is to say, the largest attribute used to recognize a feature of interest can be at most 30 times larger than the smallest. For example, it is readily possible to discriminate between membranes associated with 'darker vesicles' from the membrane of light-colored vesicles in platelets (**Fig. 2c**). However, annotating regions inside the 'darker vesicles' works poorly. This is because to annotate regions inside those vesicles the two aspects of this feature are 'high intensity value' and 'enclosed by membrane', and at the center of a large vesicle', the distance to the nearest membrane may be more than 30 times larger than the thickness of the membrane, so the neural network will have difficulty recognizing the two aspects at the same time, which often leaves the center of vesicles empty. This does not prevent the entire set of CNNs from accurately discriminating both large and small objects, since the scale can be set differently for each CNN by prescaling the input tomogram.

Data source. The PC12 cells were obtained from L. Thompson from UC Irvine[29] and cultured as previously described. These cells were intended for an overexpression experiment not described here, so these PC12 cells may be slightly perturbed from 'native'. This has no impact on our use of the cells here as a target for annotation. Cells in the log phase of growth were replated on collagen coated R2/2 Gold Quantifoil finder grids (Quantifoil, Germany) and grown overnight. A 3 µl droplet of 15 nm gold fiducial markers was applied to the cell grids right before they were plunge frozen using a Leica EMGP grid plunger (Leica, Inc.). Tilt series were collected on a JEOL JEM-2100 microscope at 15,000× magnification with a targeted defocus of –8 µm on a Gatan 4,000 × 4,000 CCD camera (Gatan, Inc.). The sampling of the data was calibrated to be 7.15 Å/pixel. A tilt series consisted of ~40 images of the specimen tilted from –60° to +60° with a 3° step increment. The accumulated exposure for each tilt series was ~60 electrons/Å^2. Alignment and 3D tomographic reconstruction were performed using IMOD.

The procyclic form of *Trypanosoma brucei* cell line 29.13[30] was cultivated in Cunningham's medium[31] supplemented with 15% heat-inactivated tetracycline-free FBS (Clontech Laboratories, CA), 15 µg/ml G418, and 50 µg/ml hygromycin at 28 °C. For vitrification, 4 µl cell samples containing 25 nm gold fiducial were

doi:10.1038/nmeth.4405

loaded on a lacey carbon film copper EM grid and rapidly fixed by plunging into liquid ethane with Vitrobot Mark IV (FEI Company Ltd., Eindhoven, the Netherlands) for CryoEM observation. The tomogram was collected on a JEOL JEM2200FS microscope with a DirectElectron DE12 detector at magnification of 11,000×, equivalent to 5.3 Å/pixel.

Platelet and cyanobacteria tomograms used in this paper were previously published[13,14]. Imaging conditions and training set sizes are given in **Supplementary Table 1**.

Code availability. The set of annotation utilities is freely available as part of the open-source package EMAN2.2. Binary downloads are available at http://www.EMAN2.org, and source code is available through Github (https://github.com/cryoem/eman2). A tutorial is available at http://www.EMAN2.org, as a **Supplementary Protocol**, and via *Protocol Exchange*[32].

Data availability statement. Subtomogram averaging results and one of the full cell annotations are deposited in EMDatabank. EMD-8589, subtomogram average of microtubules from *Trypanosoma brucei*; EMD-8590, subtomogram average of ribosome from *Trypanosoma brucei*; EMD-8591, subtomogram average of protein complexes on the thylakoid membrane in *Cyanobacteria*; EMD-8592, subtomogram average of microtubules from PC12 cell; EMD-8593, subtomogram average of microtubules from human platelet; EMD-8594, automated tomogram annotation of PC12 cell.

A **Life Sciences Reporting Summary** for this paper is available.

21. Nair, V. & Hinton, G.E. In *Proc. 27th Int. Conf. Mach. Learn.* (eds. Fürnkranz, J. & Joachims, T.) 807–814 (ICML, 2010).
22. Vincent, P., Larochelle, H., Bengio, Y. & Manzagol, P.-A. In *Proc. 25th Int. Conf. Mach. Learn* (eds. McCallum, A. & Roweis, S.) 1096–1103 (ICML, 2008).
23. Hinton, G.E., Srivastava, N., Krizhevsky, A., Sutskever, I. & Salakhutdinov, R.R. Preprint at http://arxiv.org/abs/1207.0580 (2012).
24. Dieleman, S., Willett, K.W. & Dambre, J. *Mon. Not. R. Astron. Soc.* **450**, 1441–1459 (2015).
25. Zhou, J. & Troyanskaya, O.G. *Nat. Methods* **12**, 931–934 (2015).
26. Noh, H., Hong, S. & Han, B. Preprint at http://arxiv.org/abs/1505.04366 (2015).
27. Krizhevsky, A., Sutskever, I. & Hinton, G.E. *In Advances in Neural Information Processing Systems* 25 (eds. Pereira, F., Burges, C.J.C., Bottou, L. & Weinberger, K.Q.) 1097–1105 (Curran Associates, 2012).
28. Dai, W. *et al. Nat. Protoc.* **9**, 2630–2642 (2014).
29. Apostol, B.L. *et al. Proc. Natl. Acad. Sci. USA* **100**, 5950–5955 (2003).
30. Wirtz, E., Leal, S., Ochatt, C. & Cross, G.A. *Mol. Biochem. Parasitol.* **99**, 89–101 (1999).
31. Kaminsky, R., Beaudoin, E. & Cunningham, I. *Acta Trop.* **45**, 33–43 (1988).
32. Chen, M. *et al. Protocol Exchange* http://dx.doi.org/10.1038/nprot.2017.095 (2017).

nature research

Corresponding author(s): Steven J. Ludtke

☐ Initial submission ☐ Revised version ☒ Final submission

Life Sciences Reporting Summary

Nature Research wishes to improve the reproducibility of the work that we publish. This form is intended for publication with all accepted life science papers and provides structure for consistency and transparency in reporting. Every life science submission will use this form; some list items might not apply to an individual manuscript, but all fields must be completed for clarity.

For further information on the points included in this form, see Reporting Life Sciences Research. For further information on Nature Research policies, including our data availability policy, see Authors & Referees and the Editorial Policy Checklist.

Experimental design

1. Sample size

Describe how sample size was determined. N/A

2. Data exclusions

Describe any data exclusions. Particle exclusion: Online Methods, "Particle extraction and averaging" section

3. Replication

Describe whether the experimental findings were reliably reproduced. Reusability: Online Methods, "Reusability" section

4. Randomization

Describe how samples/organisms/participants were allocated into experimental groups. N/A

5. Blinding

Describe whether the investigators were blinded to group allocation during data collection and/or analysis. N/A

Note: all studies involving animals and/or human research participants must disclose whether blinding and randomization were used.

6. Statistical parameters

For all figures and tables that use statistical methods, confirm that the following items are present in relevant figure legends (or in the Methods section if additional space is needed).

n/a	Confirmed
☐	☒ The exact sample size (*n*) for each experimental group/condition, given as a discrete number and unit of measurement (animals, litters, cultures, etc.)
☒	☐ A description of how samples were collected, noting whether measurements were taken from distinct samples or whether the same sample was measured repeatedly
☒	☐ A statement indicating how many times each experiment was replicated
☒	☐ The statistical test(s) used and whether they are one- or two-sided (note: only common tests should be described solely by name; more complex techniques should be described in the Methods section)
☒	☐ A description of any assumptions or corrections, such as an adjustment for multiple comparisons
☒	☐ The test results (e.g. *P* values) given as exact values whenever possible and with confidence intervals noted
☒	☐ A clear description of statistics including central tendency (e.g. median, mean) and variation (e.g. standard deviation, interquartile range)
☒	☐ Clearly defined error bars

See the web collection on statistics for biologists for further resources and guidance.

Nature Methods: doi:10.1038/nmeth.4405

Software

Policy information about availability of computer code

7. Software

Describe the software used to analyze the data in this study.	All code used in the manuscript is available in the EMAN2 package.

For manuscripts utilizing custom algorithms or software that are central to the paper but not yet described in the published literature, software must be made available to editors and reviewers upon request. We strongly encourage code deposition in a community repository (e.g. GitHub). *Nature Methods* guidance for providing algorithms and software for publication provides further information on this topic.

Materials and reagents

Policy information about availability of materials

8. Materials availability

Indicate whether there are restrictions on availability of unique materials or if these materials are only available for distribution by a for-profit company.	N/A

9. Antibodies

Describe the antibodies used and how they were validated for use in the system under study (i.e. assay and species).	N/A

10. Eukaryotic cell lines

a. State the source of each eukaryotic cell line used.	See Online Method, "Data source" section
b. Describe the method of cell line authentication used.	N/A
c. Report whether the cell lines were tested for mycoplasma contamination.	N/A
d. If any of the cell lines used are listed in the database of commonly misidentified cell lines maintained by ICLAC, provide a scientific rationale for their use.	N/A

Animals and human research participants

Policy information about studies involving animals; when reporting animal research, follow the ARRIVE guidelines

11. Description of research animals

Provide details on animals and/or animal-derived materials used in the study.	N/A

Policy information about studies involving human research participants

12. Description of human research participants

Describe the covariate-relevant population characteristics of the human research participants.	N/A

Nature Methods: doi:10.1038/nmeth.4405