Microscopy and Microanalysis (2021), 1–8
doi:10.1017/S1431927621000386

Microscopy and Microanalysis

Original Article

Machine Learning Pipeline for Segmentation and Defect Identification from High-Resolution Transmission Electron Microscopy Data

Catherine K. Groschner[1], Christina Choi[1] and Mary C. Scott[1,2]*

[1]Department of Materials Science and Engineering, University of California Berkeley, Berkeley, CA 94720, USA and [2]Molecular Foundry, Lawrence Berkeley National Laboratory, Berkeley, CA 94720, USA

Abstract

In the field of transmission electron microscopy, data interpretation often lags behind acquisition methods, as image processing methods often have to be manually tailored to individual datasets. Machine learning offers a promising approach for fast, accurate analysis of electron microscopy data. Here, we demonstrate a flexible two-step pipeline for the analysis of high-resolution transmission electron microscopy data, which uses a U-Net for segmentation followed by a random forest for the detection of stacking faults. Our trained U-Net is able to segment nanoparticle regions from the amorphous background with a Dice coefficient of 0.8 and significantly outperforms traditional image segmentation methods. Using these segmented regions, we are then able to classify whether nanoparticles contain a visible stacking fault with 86% accuracy. We provide this adaptable pipeline as an open-source tool for the community. The combined output of the segmentation network and classifier offer a way to determine statistical distributions of features of interest, such as size, shape, and defect presence, enabling the detection of correlations between these features.

Key words: automated analysis, deep learning, HRTEM, segmentation, structure classification

(Received 14 December 2020; revised 23 February 2021; accepted 22 April 2021)

Introduction

The need for rapid, but accurate, image analysis is ubiquitous in electron microscopy studies of nanomaterials. With the advent of fast, high-efficiency electron detectors and automated imaging protocols (Schorb et al., 2019; LeBeau et al., 2020), incorporating electron microscopy into high-throughput material design efforts (Jain et al., 2013) is increasingly feasible. These new and upcoming capabilities strongly motivate automated methods to extract relevant structural features, such as nanoparticle size, shape, and defect content, from high-resolution transmission electron microscopy (HRTEM) data and to link these features to bulk properties and the study of the influence of heterogeneity on bulk behavior (Rasool et al., 2013; Orfield et al., 2015; Zhao et al., 2018). In general, protocols, which outperform classical image analysis and do not require time-consuming manual analysis, are needed. Recent advances in image interpretation using deep learning (Szegedy et al., 2013; Ronneberger et al., 2015), segmentation, via a convolutional neural net (CNN), along with other machine learning techniques, make promising routes toward the automatic interpretation of HRTEM micrographs. Here, we demonstrate a two-step pipeline to detect and classify regions of interest in HRTEM micrographs. This tool uses a CNN to identify crystalline regions (nanoparticles) from an amorphous background in the images, and then feeds individual regions of interest into a random forest classifier to detect whether or not they contain a stacking fault (Groschner et al., 2020).

Previous work has applied deep learning to atomic resolution images of nanomaterials from high-resolution scanning transmission electron microscopy and HRTEM, but this has focused on atomic column segmentation or atomic column tracking and specific structure analysis (Ziatdinov et al., 2017, 2019; Madsen et al., 2018; Schiøtz et al., 2018, 2019; Dan et al., 2019; Kalinin et al., 2019; Maksov et al., 2019; Dennler et al., 2020; Förster et al., 2020; Jany et al., 2020). Other work has focused on lower resolution TEM micrographs, where amplitude contrast is the dominant contrast mechanism (Laramy et al., 2015; Frei & Kruis, 2018; Groom et al., 2018; Güven & Oktay, 2018; Li et al., 2018; Roberts et al., 2019; Yeom et al., 2020). Less work has been done to identify isolated regions of interest in HRTEM data and classify the structures therein. Here, we aimed to first identify nanoparticle regions, isolate them, and then classify the nanoparticles according to their structure.

We have focused on conventional HRTEM as it is a dose-efficient and high frame rate imaging mode and, therefore,

*__Author for correspondence:__ Mary C. Scott, E-mail: mary.scott@berkeley.edu

Cite this article: Groschner CK, Choi C, Scott MC (2021) Machine Learning Pipeline for Segmentation and Defect Identification from High-Resolution Transmission Electron Microscopy Data. *Microsc Microanal.* doi:10.1017/S1431927621000386

© The Author(s), 2021. Published by Cambridge University Press on behalf of the Microscopy Society of America

Downloaded from https://www.cambridge.org/core. University of Colorado Boulder, on 21 May 2021 at 21:30:13, subject to the Cambridge Core terms of use, available at https://www.cambridge.org/core/terms. https://doi.org/10.1017/S1431927621000386

CrossMark

useful for fast data acquisition on a wide range of materials. However, segmentation in HRTEM is particularly challenging as the contrast between the substrate that the nanomaterial sits on and the nanomaterial itself can be very low. Because traditional image processing techniques are highly error-prone for these types of images, we have implemented a CNN for semantic segmentation, which is the segmentation of the image on a pixel-by-pixel basis. Our CNN has a modified U-Net architecture (Ronneberger et al., 2015) and can accurately segment a diverse population of nanoparticles despite a small number of training images. Segmentation is demonstrated on both gold and cadmium selenide nanoparticles. After segmentation, individual nanoparticle regions can be isolated and fed directly into existing python tools to extract size and shape statistics. To detect the presence of defects in nanoparticle regions, we implement a random forest classifier. We demonstrated the random forest classifier's ability to detect stacking faults in the CdSe subset of identified nanoparticles. Both the CNN and classifier demonstrate state-of-the-art performance at their respective tasks. While this work focuses on HRTEM images of nanoparticles supported on a carbon substrate, in principle the tool can be used to detect any regions of crystallinity in HRTEM data, and the simple random forest classifier is designed to be easily re-trained to detect a variety of types of defects, making this a flexible pipeline suited for a variety of image analysis tasks.

Methods

Strategy

The overall goal of this work was to create a tool for the automated structural analysis of nanoparticles from HRTEM data. To achieve this, we created a two-step pipeline for accurate segmentation and defect classification. The first stage of the analysis pipeline was to train a U-Net convolutional neural network to segment particle regions. The details of the architecture are shown in Figure 1. After segmentation, we isolated individual nanoparticle regions using a series of morphological closings and openings. These isolated regions were used for feature generation and training of a random forest classifier to detect stacking faults.

Data Collection and Preprocessing

We collected 46 1,024 × 1,024 micrographs of CdSe nanoparticles and 13 4,096 × 4,096 micrographs of Au nanoparticles using an aberration-corrected TEM at 300 kV. To reduce the computational demands, we sliced each micrograph into 4 or 16 512 × 512 images. To establish a ground truth for segmentation, the data were manually labeled by an experienced researcher into either a particle class or background class. The ground truth segmentation maps were created using the MATLAB labeler application (MATLAB & Simulink, 2019). All the micrographs were then processed by first applying a median filter with a 3 × 3 kernel which served to remove spurious X-rays. After filtering, the images were normalized.

Significant portions of the raw input images were carbon background, which meant that when sliced into 512 × 512 images, only 28% of images contained nanoparticles. Therefore, any image that did not contain particle class pixels was discarded from the training and test set. The remaining data were split so that 129 images were used in the training set and 43 images were in the test set. Images were rotated and flipped to augment the dataset to more than 1,000 images which were then split 4:1 into the training and validation sets.

Network Architecture

The architecture implemented was based on the U-Net architecture developed by Ronneberger et al. (2015) following the same convolution–deconvolution structure. We modified the standard U-Net structure for accurate segmentation and to prevent overfitting.

Network Training

Training was performed using the Savio GPU cluster at U.C. Berkeley using a compute node with two Intel Xeon E5-2623 v3 CPU cores and an NVIDIA K80 GPU. The node had 64 GB of available RAM. The model was built using Keras with a Tensorflow backend (Chollet, 2015–2019). Training used 129 sample images, which then fed into a Keras image augmentation generator (Chollet, 2015–2019), which randomly rotated and flipped the images. Each epoch, or complete pass through the training set, contained 1,000 samples created from rotating the original processed 129 micrographs and split between the training and validation sets in a 4:1 ratio. Training was stopped once the loss on the validation set did not decrease by 0.001 after two epochs. The model was limited to train for a maximum of ten epochs. The model used categorical cross-entropy for the loss function and Adam as the optimizer with a learning rate of 1×10^{-4} (Kingma & Ba, 2015; Goodfellow et al., 2016). Batch size was set to 20, where batch size defines how many samples from the training set are propagated through the network before updating the model weights.

Testing

A holdout test dataset of 43 images was reserved for final testing after model training had completed. The predicted particle segmentation map was thresholded to 0.5 prior to metric computation because this represented the best tradeoff between precision and recall. Precision and recall versus threshold curves for the network both on Au and CdSe data are provided in the Supplementary Materials. Reported metrics are the results from these test sets.

Alternate Methods for Segmentation

Two standard segmentation methods were applied to the dataset to compare with the U-Net segmentation. Standard thresholding was applied using Otsu's method (Otsu, 1979) to threshold out the nanoparticles based on intensity. Fourier filtering was also used to segment out crystalline regions. For this filtering, the Fourier transform of the image was multiplied by an annulus to eliminate frequencies related to the material's lattice structure. The inverse Fourier transform of the filtered image was then smoothed and thresholded to create a segmentation map.

Random Forest Classifier

After applying the neural network, the resulting segmentation masks were used to isolate particle regions. To limit the number of agglomerated regions input into the random forest, we

Downloaded from https://www.cambridge.org/core. University of Colorado Boulder, on 21 May 2021 at 21:30:13, subject to the Cambridge Core terms of use, available at https://www.cambridge.org/core/terms. https://doi.org/10.1017/S1431927621000386

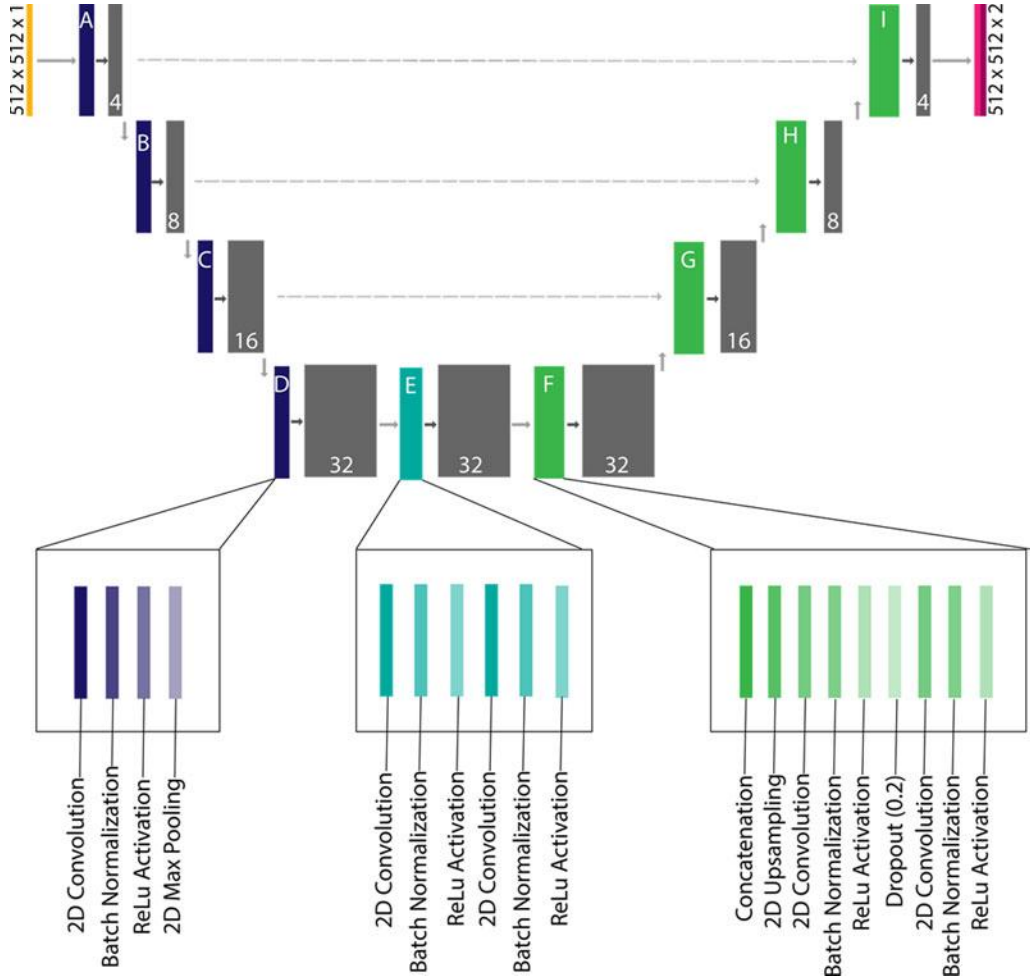

Fig. 1. The U-Net style architecture of CNN implemented for the segmentation of nanoparticle regions. Blocks colored gray represent the output features and are labeled with the number of features created. Residual blocks A–D contain a two-dimensional (2D) convolutional layer, batch normalization layer, rectified linear unit (ReLu) activation layer, and a 2D max pooling layer. Residual block E contains a 2D convolutional layer, batch normalization layer, and ReLu activation layer, which is repeated once. Residual blocks F–I contain a 2D upsampling layer, concatenation layer, a 2D convolution, batch normalization, and ReLu activation followed by a dropout layer and another set of 2D convolution, batch normalization, and ReLu activation layers. The final layer is a softmax activation layer which outputs two segmentation maps one for each class, background and particle.

implemented morphological filtering on predicted segmentation maps using the scikit-image python package (Van der Walt et al., 2014). Regions which were too large to be independent particles were discarded before stacking fault detection. The random forest was developed using the scikit-learn python package (Pedregosa et al., 2011). 329 particle regions were in the training set, and 163 regions were in the test and validation set. The training set had 71 examples of stacking faults, 156 examples of no stacking fault, 80 examples of particles not on the zone axis, 60 examples of agglomerations, and 43 empty regions. The training and validation set were manually labeled. The random forest had 500 decision trees, and the criterion for tree splitting was Gini impurity, which calculates the probability of classifying the data incorrectly. The feature set consisted of the mean, standard deviation, and center of mass of the radially integrated Fourier transform of the particle region and the mean and standard deviation of the nanoparticle image in real space. This feature set was selected such that validation accuracy and true positive rate for each class were highest. These results lead to a small and informationally dense feature set based on summary statistics.

Results and Discussion

Evaluation Metrics for Segmentation

To evaluate the neural network's ability to segment nanoparticles from the micrograph, we used three evaluation metrics: (1) Dice coefficient (DICE), also known as F1-score, (2) precision, and (3) recall. We look at three metrics because each offers a difference weighting of error types. Dice coefficient is a standard metric for segmentation tasks and measures the union of positive pixels between the true and predicted segmentation masks normalized by the total number of positive pixels in the two groups. The Dice coefficient can be a very strict metric for small particle regions because of the weighting of true positive pixels, shown in Equation (1), where TP stands for true positives, FP for false positives, and FN for false negatives (Taha & Hanbury, 2015). Fewer positive pixels overall means small errors can have a strong effect on the metric. Therefore, it is also important to consider other metrics, such as precision and recall. Precision essentially measures how successful the neural network is at only selecting true particles. Recall, on the other hand, provides information

Downloaded from https://www.cambridge.org/core. University of Colorado Boulder, on 21 May 2021 at 21:30:13, subject to the Cambridge Core terms of use, available at https://www.cambridge.org/core/terms. https://doi.org/10.1017/S1431927621000386

Table 1. Performance Metrics for Test Sets.

	DICE	Precision	Recall
Combined Dataset	0.8	0.82	0.78
CdSe Data	0.59	0.56	0.62
Au Data	0.89	0.95	0.84

on how well the network can find particles.

$$\text{DICE} = \frac{2\text{TP}}{2\text{TP} + \text{FP} + \text{FN}} \tag{1}$$

Performance of U-Net Segmentation and Comparison with Standard Image Processing Methods

We find segmentation via U-Net style CNN to be the most effective way to segment micrographs as compared with standard methods. Overall, the Dice coefficient across the test set was 0.8. Separating the test set according to sample material, the network achieved a Dice coefficient for Au nanoparticle micrographs of 0.89, while the CdSe nanoparticles were segmented with a Dice coefficient of 0.59, which is on par with other state-of-the-art segmentation procedures (DeCost et al., 2019). Further metrics are provided in Table 1 and the confusion matrices are presented in Figure 2. Sample segmentations for both are shown in Figure 3. CdSe particles clearly are the much more difficult segmentation case, both for human and computer labelers. CNN-based segmentation is likely limited due to the limited signal relative to the background and the size of the particle regions (limiting the context for the neural network to learn). We explored whether training solely on CdSe would improve the segmentation, but we found that training on a diverse set of materials actually improved results. These results are presented in the Supplementary Materials. Despite a lower Dice coefficient on the CdSe data, the network created by the demonstrated training procedure results in a network which tends to underpredict particle regions, as shown in Figure 2a, from the low false-positive rate for the particle class. This bias is preferred to the over prediction of particles which would make later classification challenging.

To determine if the network was more successful than standard image processing techniques, we segmented the micrographs using Otsu's method (Otsu, 1979) and using Fourier filtering (Buseck et al., 1988; De Jong et al., 1989). More information for each segmentation method is supplied in the "Methods" section. We found that basic thresholding on the complete CdSe test set achieved a Dice coefficient of 0.21 and Fourier filtering achieved a Dice coefficient of 0.33. We note that this is significantly worse than using Otsu's method to threshold the Au test set, which leads to a Dice coefficient of 0.45, or using Fourier filtering, which leads to a Dice coefficient of 0.78. The Dice coefficients for the combined test set was 0.33 and 0.52 for thresholding via Otsu's method and Fourier filtering, respectively. This takes into account optimizing the radius of the annulus mask in Fourier filtering, which impacts the segmentation quality. Sample segmentations for each method are shown in Figure 4. Clearly, on all accounts, the neural network performs significantly better than standard segmentation methods, but it is a particular improvement for the CdSe dataset, where there is little amplitude contrast. In addition, beyond the improved segmentation, using a neural network also has the benefit of not hand tuning parameters for each image, which are critical in standard processing methods. Therefore, for this type of TEM data, U-Net provides a better, and higher throughput, means of particle segmentation.

CdSe Particle Statistics from Random Forest Classification

The primary reason to develop a highly accurate segmentation technique for HRTEM is to enable the high-throughput analysis of crystal structures. Segmented regions of interest can be used in further classification models and automated statistics. Statistics on size and shape can be provided directly from segmentation results. Segmented regions can also be passed to classifiers for structure or defect detection as we will demonstrate.

Stacking faults in CdSe are of interest due to the influence they can have on particle growth and, therefore, shape of CdSe nanoparticles (Peng et al., 2000). Here, we show that a simple random forest can be used to detect stacking faults. A random forest was chosen for several reasons, including the excellent performance of ensemble methods as well as the speed and simplicity to train (Geron, 2017). Combined with high model interpretability, makes this classifier a good choice for fast iteration and rapid development.

The random forest demonstrated is used to categorize the CdSe nanoparticles into five different classes: (1) particle has a stacking fault visible, (2) particle does not have stacking fault

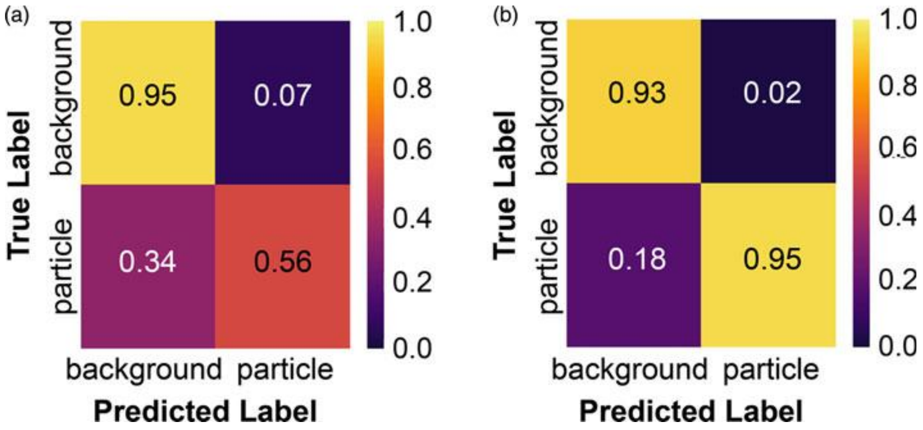

Fig. 2. The confusion matrices for the trained U-Net on (**a**) CdSe test set and (**b**) for the Au test set.

Downloaded from https://www.cambridge.org/core. University of Colorado Boulder, on 21 May 2021 at 21:30:13, subject to the Cambridge Core terms of use, available at https://www.cambridge.org/core/terms. https://doi.org/10.1017/S1431927621000386

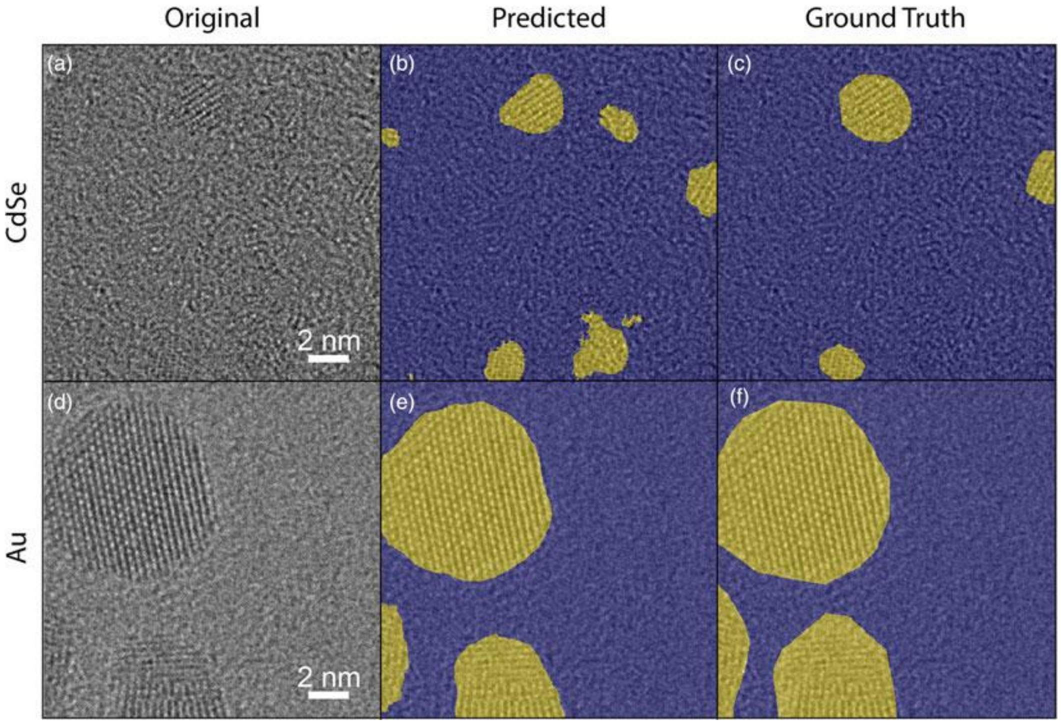

Fig. 3. (**a–c**) a sample micrograph, segmentation from the network, and ground truth segmentation from the CdSe dataset. (**d–f**) a sample micrograph, segmentation from the network, and ground truth segmentation from the Au dataset. For all segmentation maps, yellow represents regions predicted to be in the particle class, and blue represents regions predicted as background.

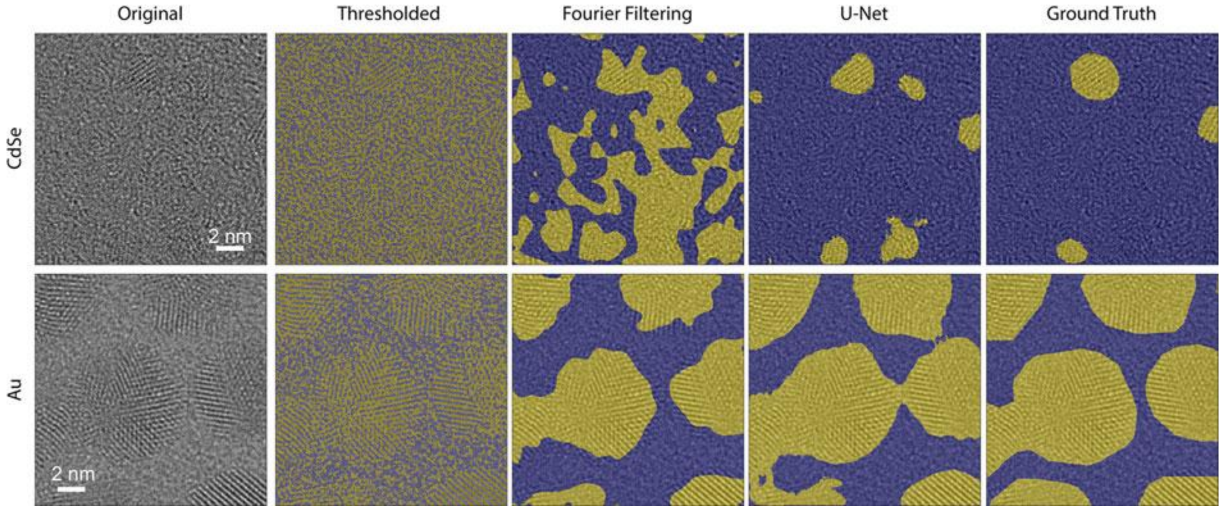

Fig. 4. Sample micrographs of CdSe and Au particles and the resulting segmentation maps when segmented by Otsu's method, Fourier filtering, and U-Net. For all segmentation maps, yellow represents regions predicted to be in the particle class, and blue represents regions predicted as background.

visible, (3) particle does not have atomic column contrast (misoriented), (4) region is an agglomeration of particles, and (5) region is background. Sample micrographs for each class are shown in Figures 5a–5e. We included the last category to handle the few cases in which the neural network segments a region incorrectly since perfect segmentation cannot be guaranteed from our network. The feature set we developed for the random forest is extremely simple and consists only of the mean, standard deviation, and center of mass of the radially integrated Fourier transform of the particle region and the mean and standard deviation of the nanoparticle image in real space. Figure 5f shows the confusion matrix for the random forest classifier on the test dataset. The confusion matrix provides the fraction of correct and incorrect labels for each class. The random forest

Downloaded from https://www.cambridge.org/core. University of Colorado Boulder, on 21 May 2021 at 21:30:13, subject to the Cambridge Core terms of use, available at https://www.cambridge.org/core/terms. https://doi.org/10.1017/S1431927621000386

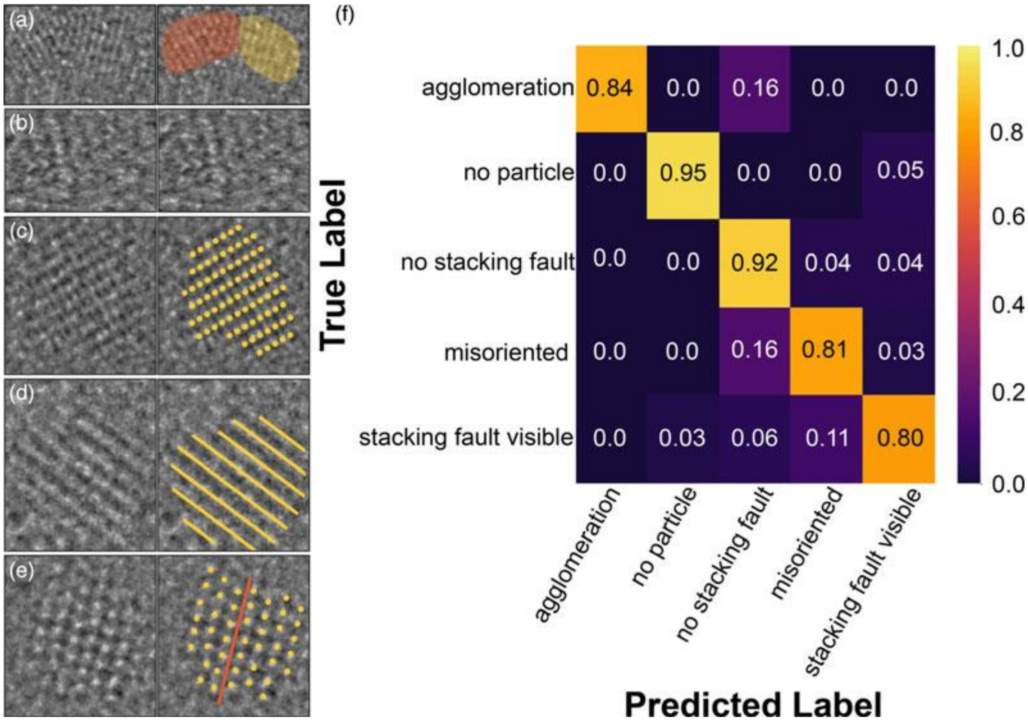

Fig. 5. Sample micrographs, original on the left and annotated on the right, of the five classes predicted: (**a**) agglomeration (the two particle regions are highlighted in orange and yellow, respectively), (**b**) no particle, (**c**) no stacking fault (atomic columns, marked in yellow, are aligned), (**d**) misoriented (atomic planes, marked by yellow lines, are visible but atomic columns are not), and (**e**) stacking fault visible (atomic columns, marked in yellow, can be seen to be offset across the stacking fault, marked in orange). (**f**) Confusion matrix for the random forest classifier.

achieves 86% class balanced accuracy compared with the expert labels given. The random forest predicted that 52% of particles were in the correct orientation to observe a stacking fault, and of those 38% of particles contain a stacking fault. This can be compared with the ground truth of 49% of particles which were in the correct orientation to observe a stacking fault and the 35% of those particles that contained a stacking fault. As a consequence, we can see that using an extremely limited feature space still yields good classification of particles.

In summary, this two-step pipeline offers a flexible method to identify crystalline regions of interest and classify individual regions according to known features. The segmentation portion of the pipeline outperforms standard segmentation methods, even when those methods are optimized for individual datasets. The easily retrainable random forest classifier enables the detection of known features within the identified regions. We note that the outputted individual regions could also be input into unsurpervised classifiers for the detection of previously unidentified features. Further work could also aim to implement more complex structure and defect classification methods (Li et al., 2018; Maksov et al., 2019). One particularly useful future area of work would be to develop zone axis classification alongside defect identification, which would aid in the interpretation of observed defects.

Coupled with existing size and shape analysis tools, the output of the segmentation network and classifier demonstrated here offer a way to determine statistical distributions of features of interest, such as size, shape, and defect presence, and importantly, allow one to detect correlations between these features. This capability will be critical to offer insight into nanomaterial population evolution during high-throughput synthesis studies or to identify key structural features that influence bulk properties in combinatorial synthesis efforts.

Conclusion

Here, we have demonstrated a method of flexible automated analysis for nanoparticles in HRTEM by combining a neural network for segmentation and a random forest for classification tasks. We have shown that a U-Net architecture can far outperform traditional image processing techniques for the segmentation of HRTEM images, while a simple classification tool can accurately classify structural features using a very limited number of parameters. Breaking apart the segmentation and classification tasks leads to an accurate tool with limited data labeling and feature engineering requirements. Moreover, this pipeline provides a flexible open-source tool as the base for further analysis and classification tools for local atomic structure. Such quantitative, automated analysis will have significant implications for a broad range of nanoparticle synthesis, structure–property relationship, and combinatorial synthesis studies.

Availability of data. The complete workflow for micrograph preprocessing, network training, and testing is available in Jupyter notebooks at https://github.com/ScottLabUCB/HTTEM/tree/master/pyNanoFind. All micrographs and ground truth segmentation maps are available on Zenodo: https://doi.org/10.1017/S1431927621000386.

Supplementary material. To view supplementary material for this article, please visit https://doi.org/10.1017/S0885715621000324.

Acknowledgments. The authors thank Haoran Yang and Emory Chan for providing the CdSe nanoparticles. The authors also thank Colin Ophus,

Downloaded from https://www.cambridge.org/core. University of Colorado Boulder, on 21 May 2021 at 21:30:13, subject to the Cambridge Core terms of use, available at https://www.cambridge.org/core/terms. https://doi.org/10.1017/S1431927621000386

Mike MacNeil, and Tess Smidt for their feedback and comments during the writing of this manuscript.

Author contributions statement. M.C.S. and C.K.G. conceived the project and designed the experiments. C.K.G. and C.C. optimized the network architecture. C.K.G. acquired the training images, trained the network, and tested the network. C.K.G. and M.C.S. wrote the manuscript. All authors discussed the results and commented on the manuscript.

Financial support. Work at the Molecular Foundry was supported by the Office of Science, Office of Basic Energy Sciences, of the U.S. Department of Energy under Contract No. DE-AC02-05CH11231. This material is based upon work supported by the National Science Foundation Graduate Research Fellowship under Grant No. DGE-1752814. This work was also supported by National Science Foundation STROBE grant DMR-1548924. CdSe synthesis was supported by the Department of Energy, Office of Energy Efficiency and Renewable Energy (EERE), under Award Number DE-EE0007628.

Conflict of interest. The authors declare no competing financial or non-financial interests.

References

Buseck PR, Epelboin Y & Rimsky A (1988). Signal processing of high-resolution transmission electron microscope images using Fourier transforms. *Acta Cryst* **A44**, 975–986.

Chollet F (2015–2019). Keras. Available at https://keras.io/.

Dan J, Zhao X & Pennycook SJ (2019). A machine perspective of atomic defects in scanning transmission electron microscopy. *InfoMat* **1**(3), 359–375. doi:10.1002/inf2.12026.

DeCost BL, Francis T & Holm E (2019). High throughput quantitative metallography for complex microstructures using deep learning: A case study in ultrahigh carbon steel. *Microsc Microanal* **25**(1), 21–29. doi:10.1017/S1431927618015635.

De Jong AF, Coene W & Van Dyck D (1989). Image processing of HRTEM images with non-periodic features. *Ultramicroscopy* **27**(1), 53–65. doi:10.1016/ 0304-3991(89)90200-3.

Dennler N, Foncubierta-Rodriguez A, Neupert T & Sousa M ((2020). Learning-based defect recognition for quasi-periodic microscope images,.

Förster GD, Castan A, Loiseau A, Nelayah J, Alloyeau D, Fossard F, Bichara C & Amara H (2020). A deep learning approach for determining the chiral indices of carbon nanotubes from high-resolution transmission electron microscopy images. *Carbon* **169**, 465–474. doi:10.1016/j.carbon.2020.06.086. arXiv: 2003.11505.

Frei M & Kruis FE (2018). Fully automated primary particle size analysis of agglomerates on transmission electron microscopy images via artificial neural networks. *Powder Technol* **332**, 120–130. doi:10.1016/j.powtec.2018.03.032. arXiv: 1806.04010.

Geron A (2017). *Hands-On Machine Learning with Scikit-Learn & TensorFlow*, 1st ed. Sebatopol: O'Reilly Media.

Goodfellow I, Bengio Y & Courville A (2016). *Deep Learning*. MIT Press. Available at http://www.deeplearningbook.org.

Groom DJ, Yu K, Rasouli S, Polarinakis J, Bovik AC & Ferreira PJ (2018). Automatic segmentation of inorganic nanoparticles in BF TEM micrographs. *Ultramicroscopy* **194**, 25–34. doi:10.1016/j.ultramic.2018.06.002.

Groschner CK, Choi C & Scott MC (2020). Methodologies for successful segmentation of HRTEM images via neural network. *arXiv*: 2001.05022. Available at http://arxiv.org/abs/2001.05022.

Güven G & Oktay AB (2018). Nanoparticle detection from TEM images with deep learning. In *26th IEEE Signal Processing and Communications Applications Conference*, SIU 2018, pp. 1–4.

Jain A, Ong SP, Hautier G, Chen W, Richards WD, Dacek S, Cholia S, Gunter D, Skinner D, Ceder G & Persson KA (2013). Commentary: The materials project: A materials genome approach to accelerating materials innovation. *APL Mater* **1**(1), 011002. doi:10.1063/1.4812323.

Jany BR, Janas A & Krok F (2020). Automatic microscopic image analysis by moving window local Fourier transform and machine learning. *Micron* **130**, 102800. doi:10.1016/j.micron.2019.102800.

Kalinin SV, Lupini AR, Dyck O, Jesse S, Ziatdinov M & Vasudevan RK (2019). Lab on a beam—Big data and artificial intelligence in scanning transmission electron microscopy. *MRS Bulletin* **44**(7), 12742–12752. doi:10.1557/mrs.2019.159.

Kingma DP & Ba JL (2015). Adam: A Method for Stochastic Optimization, pp. 1–15. arXiv: 1412.6980v9.

Laramy CR (2015). High-throughput, algorithmic determination of nanoparticle structure from electron microscopy images. *ACS Nano* **9**(12), 12488–12495. doi:10.1021/acsnano.5b05968.

LeBeau J, Kumar A & Hauwiller M (2020). A universal scripting engine for transmission electron microscopy. *Microsc Microanal* **26**(S2), 2958–2959. doi:10.1017/S1431927620023338.

Li W, Field KG & Morgan D (2018). Automated defect analysis in electron microscopic images. *npj Comput Mater* **4**(1), 1–9. doi:10.1038/ s41524-018-0093-8.

Madsen J, Liu P, Kling J, Wagner JB, Hansen TW, Winther O & Schiøtz J (2018). A deep learning approach to identify local structures in atomic-resolution transmission electron microscopy images. *Adv Theory Simul* **1**, 8. doi:10.1002/adts.201800037.

Maksov A, Dyck O, Wang K, Xiao K, Geohegan DB, Sumpter BG, Vasudevan RK, Jesse S, Kalinin SV & Ziatdinov M (2019). Deep learning analysis of defect and phase evolution during electron beam-induced transformations in WS2. *npj Comput Mater* **5**(1), 12. doi:10. 1038/ s41524-019-0152-9.

MATLAB & Simulink (2019). Get Started with the Image Labeler. Available at https://www.mathworks.com/help/vision/ug/get-started-with-the-image-labeler.html (retrieved May 30, 2019).

Orfield NJ, Mcbride JR, Keene JD & Davis LM (2015). Correlation of atomic structure and photoluminescence of the same quantum dot: Pinpointing surface and internal defects that inhibit photoluminescence. *ACS Nano* **9** (1), 831–839. doi:10.1021/nn506420w.

Otsu N (1979). A threshold selection method from gray-level histograms. *IEEE Trans Syst, Man, and Cybern SMC* **9**(1), 62–66. doi:10.1109/ TSMC.1979.4310076.

Pedregosa F, Varoquaux G, Gramfort A, Michel V, Thirion B, Grisel O, Blondel M, Prettenhofer P, Weiss R, Dubourg V, Vanderplas J, Passos A, Cournapeau D, Brucher M, Perrot M & Duchesnay E (2011). scikit-learn: Machine learning in python. *J Mach Lear Res* **12**, 2825–2830.

Peng X, Manna L, Yang W, Wickham J, Scher E, Kadavanich A & Alivisatos AP (2000). Shape control of CdSe nanocrystals. *Nature* **404**(6773), 59–61. doi:10.1038/35003535.

Rasool HI, Ophus C, Klug WS, Zettle A & Gimzewski JK (2013). Measurement of the intrinsic strength of crystalline and polycrystalline graphene. *Nat Commun* **4**, 1. doi:10.1038/ncomms3811.

Roberts G, Haile SY, Sainju R, Edwards DJ, Hutchinson B & Zhu Y (2019). Deep learning for semantic segmentation of defects in advanced STEM images of steels. *Sci Rep* **9**(1), 1–12. doi:10.1038/s41598-019-49105-0.

Ronneberger O, Fischer P & Brox T (2015). U-Net: Convolutional networks for biomedical image segmentation. In *Medical Image Computing and Computer-Assisted Intervention–MICCAI 2015*, Navab N (Ed.), Vol. **9351**, pp. 234–241. Cham: Springer International Publishing. doi:10.1007/ 978-3-319-24574-4_28.

Schiøtz J, Madsen J, Liu P, Winther O, Kling J, Wagner J & Hansen T (2018). Identifying atoms in high resolution transmission electron micrographs using a deep convolutional neural net. *Microsc Microanal* **24**(S1), 512–513. doi:10.1017/S1431927618003057.

Schiøtz J, Madsen J, Østergaard BJG, Dreisig AS, Liu P, Helveg S, Winther O, Kling J, Wagner JB & Hansen TW (2019). Using neural networks to identify atoms in HRTEM images. *Microsc Microanal* **25**(S2), 216–217. doi:10.1017/S1431927619001818.

Schorb M, Haberbosch I, Wim JHH, Schwab Y & Mastronarde DN (2019). Software tools for automated transmission electron microscopy. *Nat Methods*. doi:10.1038/s41592-019-0396-9.

Downloaded from https://www.cambridge.org/core. University of Colorado Boulder, on 21 May 2021 at 21:30:13, subject to the Cambridge Core terms of use, available at https://www.cambridge.org/core/terms. https://doi.org/10.1017/S1431927621000386

Szegedy C, Toshev A & Erhan D (2013). Deep neural networks for object detection. In *Advances in Neural Information Processing Systems 26*, Burges CJC (Ed.), pp. 2553–2561. Lake Tahoe: Curran Associates, Inc.

Taha AA & Hanbury A (2015). Metrics for evaluating 3D medical image segmentation: Analysis, selection, and tool. *BMC Med Imaging* **15**(1). doi:10.1186/s12880-015-0068-x.

Van der Walt S, Schönberger JL, Nunez-Iglesias J, Boulogne F, Warner JD, Yagner N, Gouillart E & Yu T (2014). scikit-image: Image processing in python. *PeerJ* **2**, e453. doi:10.7717/peerj.453.

Yeom J, Stan T, Hong S & Voorhees PW (2020). Segmentation of experimental datasets via convolutional neural networks trained on phase field simulations. *SSRN Electron J*. doi:0.2139/ssrn.3680400.

Zhao S, Austin N, Li M, Song Y, House SD, Bernhard S, Yang JC, Mpourmpakis G & Jin R (2018). Influence of atomic-level morphology on catalysis: The case of sphere and rod-like gold nanoclusters for CO_2 electroreduction. *ACS Catal* **8**(6), 4996–5001. doi:10.1021/acscatal.8b00365.

Ziatdinov MA, Dyck O, Jesse S & Kalinin SV (2019). Atomic mechanisms for the Si atom dynamics in graphene: Chemical transformations at the edge and in the bulk.

Ziatdinov M, Dyck O, Maksov A, Li X, Sang X & Xiao K (2017). Deep learning of atomically resolved scanning transmission electron microscopy images: Chemical identification and tracking local transformations. *ACS Nano* **11**(12), 12742–12752. doi:10.1021/acsnano.7b07504.

Downloaded from https://www.cambridge.org/core. University of Colorado Boulder, on 21 May 2021 at 21:30:13, subject to the Cambridge Core terms of use, available at https://www.cambridge.org/core/terms. https://doi.org/10.1017/S1431927621000386