PNNL-33751

Development of Multimodal Few-Shot Analytics for Electron Micrographs

December 2022

1 Arman H. Ter-Petrosyan
2 Jenna Pope
3 Christina M. Doty
4 Bethany E. Matthews
5 Sarah M. Akers
6 Steven R. Spurgeon

PACIFIC NORTHWEST NATIONAL LABORATORY
operated by
BATTELLE
for the
UNITED STATES DEPARTMENT OF ENERGY
under Contract DE-AC05-76RL01830

Printed in the United States of America

Available to DOE and DOE contractors from
the Office of Scientific and Technical Information,
P.O. Box 62, Oak Ridge, TN 37831-0062
www.osti.gov
ph: (865) 576-8401
fax: (865) 576-5728
email: reports@osti.gov

Available to the public from the National Technical Information Service
5301 Shawnee Rd., Alexandria, VA 22312
ph: (800) 553-NTIS (6847)
or (703) 605-6000
email: info@ntis.gov
Online ordering: http://www.ntis.gov

PNNL-33751

Development of Multimodal Few-Shot Analytics for Electron Micrographs

December 2022

1 Arman H. Ter-Petrosyan
2 Jenna Pope
3 Christina M. Doty
4 Bethany E. Matthews
5 Sarah M. Akers
6 Steven R. Spurgeon

Prepared for
the U.S. Department of Energy
under Contract DE-AC05-76RL01830

Pacific Northwest National Laboratory
Richland, Washington 99354

Abstract

Recent advances in materials data analytics have provided new avenues for determining process-structure-property (PSP) linkages in a variety of materials. Machine learning techniques including few-shot learning have increased the efficiency of classifying microscopy images for the purposes of material characterization. Attempts at creating a multimodal approach can provide further improvements to current models and help extract more salient features from data. In this vein, raw spectrum data was taken to provide an additional modality to our current pyCHIP classifier. Modifications in segmentation also show potential in improving the accuracy of the pyCHIP classifier. Classifier output was analyzed using network graphs and unsupervised clustering algorithms such as spectral clustering to detect better segmentation methods than the current “chipping” approach. We suggest that the chip selection process can be automated in the future using a combination of these techniques to enable high-throughput analyses.

Acknowledgments

A.H.T. was supported in part by the U.S. Department of Energy, Office of Science, Office of Workforce Development for Teachers and Scientists (WDTS) under the Science Undergraduate Laboratory Internships Program (SULI). This research was supported by a Chemical Dynamics Initiative (CDi) Laboratory Directed Research and Development (LDRD) project at Pacific Northwest National Laboratory (PNNL). PNNL is a multiprogram national laboratory operated for the U.S. Department of Energy (DOE) by Battelle Memorial Institute under Contract No. DE-AC05-76RL0-1830.

A big thank you to Steven Spurgeon, Jenna Pope, Christina Doty, Bethany Matthews, and Sarah Akers at PNNL. Steven was gracious enough to make time in his busy schedule to work through any problems I came across throughout my internship, ensuring that my research experience was both productive and informative. Jenna introduced me to NetworkX and regularly helped me write code for my project. Christina introduced me to the research team's codebase, helped me setup my computing environment for the project, and addressed any issues regarding version control. Bethany provided spectral data and introduced me to Pathfinder. Sarah addressed questions I had regarding the multimodal model discussed in the paper.

Also, thank you to Stephen Seddio at Thermo Fisher for answering my endless questions about Pathfinder. Without Steve's help, extracting the data needed to develop the multimodal model discussed in this paper would not have been possible.

Contents

Figures

1.0 Introduction

A longstanding objective of materials science research is the discovery of process-structure-property (PSP) linkages. The application of data science techniques to materials data allows for increased efficiency in mining and extracting of valuable materials knowledge. (Kalidindi & De Graef, 2015) This materials knowledge can accelerate the development of novel materials for a variety of modern applications including energy storage, quantum computing, and transformative manufacturing. (Olszta , et al., 2022)

One possible avenue is autonomous experimentation. Integrating artificial intelligence (AI) and machine learning (ML) into current microscopy setups allows researchers to conduct high-throughput analyses. Olszta et al. demonstrated the use of few-shot ML feature classification in a scanning transmission electron microscope (STEM) to automatically identify desired features in the image data depending on the task at hand. (Olszta , et al., 2022) Classification algorithms can be used to analyze the microstructure of materials. Akers et al. have developed a few-shot learning implementation that offers experimenters the ability to both rapidly and accurately classify microscopy images taken by microscopes using a small training set. (Akers, et al., 2021) They demonstrated that… the algorithm does whatever… The pyCHIP classifier is one such model that implements this functionality and has proven to be useful in classifying microscopy images of several different kinds of materials. (Doty, et al., 2022) We believe that extending the model to accept other types of data such as spectral data can improve accuracy and extract more salient features from materials data.

2.0 Results and Discussion

The current iteration of the pyCHIP classifier allows researchers to segment microscopy images into chips of equal dimension. Though straightforward, this approach has proven to be useful accurately classifying the composition of a material's microstructure. Other segmentation techniques can provide additional benefits besides improved accuracy such as feature detection. Specifically, unsupervised machine learning models have been used for the segmentation of static and dynamic atomic-resolution microscopy data sets in the form of images and video sequences. (Wang, Freysoldt, Zhang, Liebscher, & Neugebauer, 2021) Models like spectral clustering help detect similarities in data and are simple to implement. Many network graphing packages include clustering or community detection capabilities and are great for visualization purposes.

2.1 Image Preprocessing

Image preprocessing plays a role in the outcome of the analysis described above and so it is important to consider the type of preprocessing used when running our analysis. Since earlier iterations were not picking up many features present in the image, we decided to update our PychipClassifier preprocessing method to include an option which normalizes tensor images with mean standard deviation using a PyTorch normalization method. This helps ensure all image data follow similar distributions.

Using each of the three preprocessing options available, chipped images were colored based on the labels predicted by the classifier to help visualize the role the different preprocessing procedures playing in classifier outcomes. The classifier was trained and tested on a microscopy image of SrTiO3/Ge. Results are presented in the figure below.

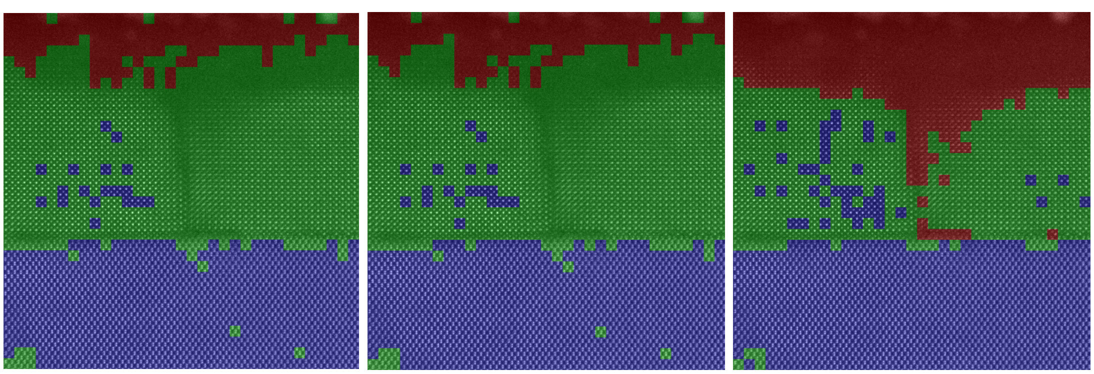

Figure 3. Chipped images colored based on predicted labels when classifier uses (from left to right) (a) no preprocessing, (b) PyTorch normalization, and (c) CLAHE preprocessing. Note that images (a) and (b) are identical, there is no discernable difference is when using PyTorch normalization on the SrTiO3/Ge image.

2.2 Unsupervised Classification

To quantify the similarity between chips in segmented microscopy images, we compute the p-norm distance between encodings of all pairs of chips in the image. A PychipClassifier method cleverly titled chip2chip was made to do this which took several parameters: encoder, maximum query size, seed, save path, filename, and cutoff. The encoder parameter can be set to one of seven PyTorch encoders used by the pyCHIP classifier: resnet18, resnet34, resnet50, resnet101, resnet152, torch101, and shufflenet. Each of these encoders were pretrained on the ImageNet database of images. The cutoff parameter can be set to any positive rational value and represents the acceptable upper limit for classifier output. Thus, chips with classifier output less than the cutoff are used when computing the distances. Once computed, the data is organized in a data frame with columns representing the chips, distances, and normalized distances. A summary of the steps implemented by the chip2chip method is given below.

1. Load the pretrained encoder
2. Chip the image
3. Obtain encodings of each chip
4. Compute the p-norm distance between each pair of chips
5. Return normalized distances in a dataframe

Histograms were made to visualize the distribution of distances between all possible chip pairs in a microscopy image of $SrTiO_3$/Ge. The distances were computed using the chip2chip method described above. Since the pyCHIP classifier allows for seven possible encodings, a histogram was made for each one. The image below shows similarity histograms computed using resnet18, resnet34, resnet50, or shufflenet as the encoder.

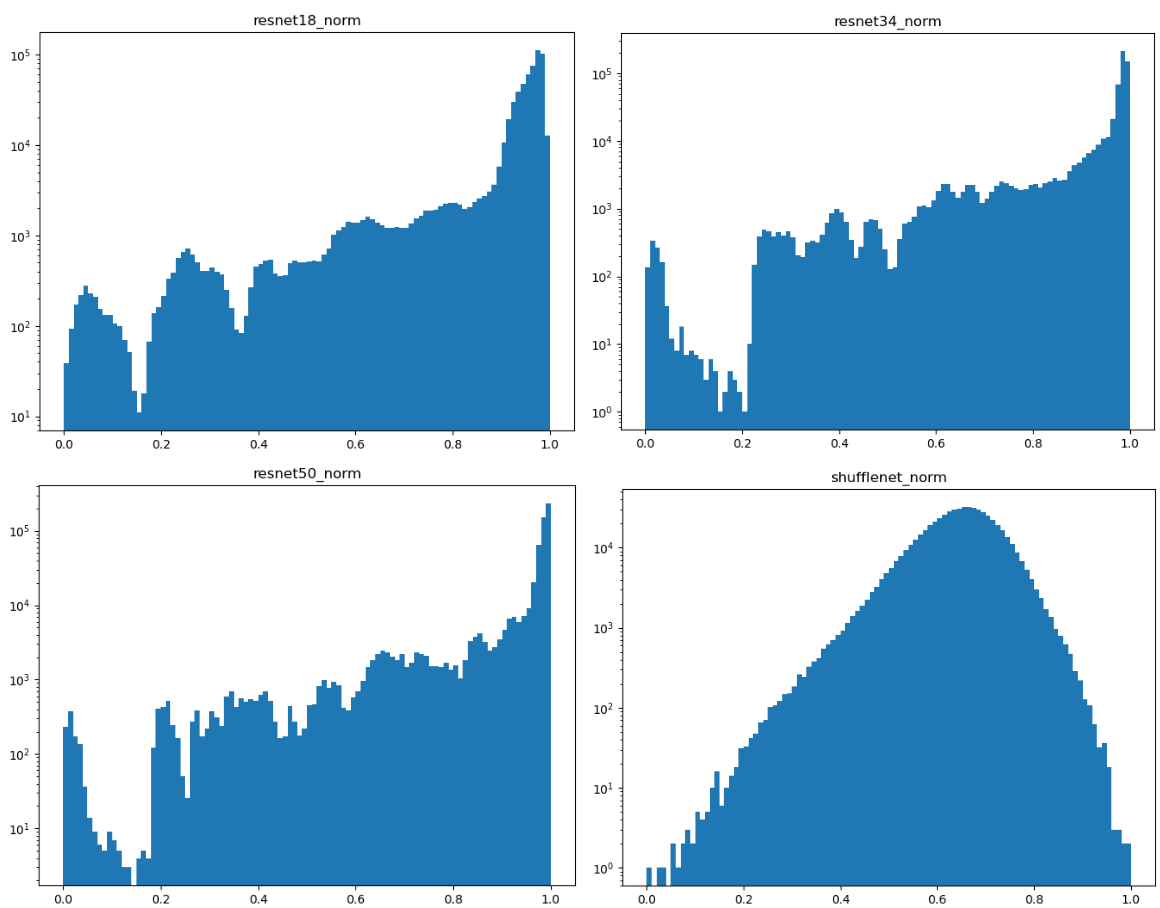

Figure 4. Histograms displaying the distribution of distances between all possible chip pairs when using (clockwise from top left) (a) resnet18, (b) resnet34, (c) resnet50, and (d)

shufflenet encodings. PyTorch normalization applied for preprocessing the original image. The distance used is scaled to be in the unit interval where 0 is no similarity and 1 is high similarity. Note that the y-axis is logarithmic.

Distributions that are nonuniform show that some pairs of chips tend to be more similar and others less. This is a sign that some chips potentially share similar features and characteristics that are being picked up by the particular combination of preprocessing steps and encoder used. Although the histograms above show that none of these encodings are particularly suited for picking up differences that could be exploited for segmentation, it is still likely that some encodings are better than others.

2.3 Network Graph Analysis

Using the output of chip2chip, a network graph is constructed using Python's NetworkX package. The nodes of the network are the chips and edges are drawn connecting all nodes. Since shorter distances correspond to greater similarity and farther distances correspond to less similarity, the normalized distances are subtracted from one to provide the appropriate weights used for building the network graph. An example of a network graph is shown in the figure below.

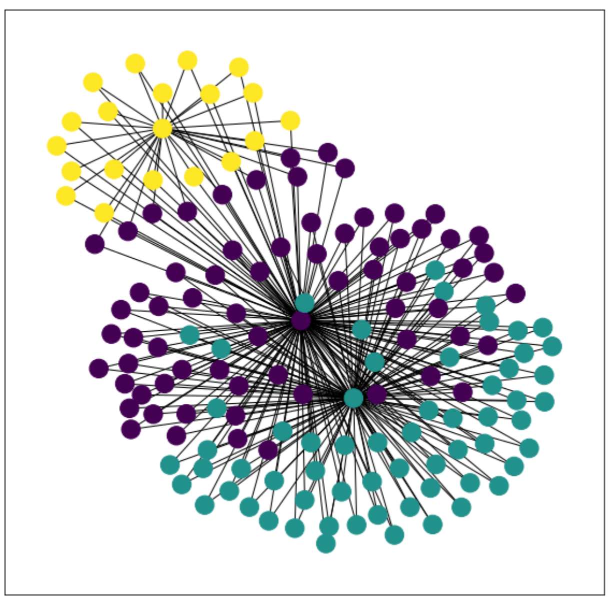

Figure 3. Network graph with a NetworkX community detection tool applied on top showing three distinct communities in the data. Each node in the graph represents a chip from the SrTiO3/Ge microscopy image (only a sample of chips was used in constructing this graph). Edges between nodes were weighed by the scores they received for each of the three possible classes.

Using NetworkX's implementation of the Louvain method for community detection, we can detect the communities in our network graph. The chips in the image can then be colored according to the community they belong to using another method we devised to help better visualize where the communities are located throughout the image. This can give us a sense as to whether the community detection algorithm is picking up on any discernable features present in the image. The outcome of this analysis is presented in the figure below.

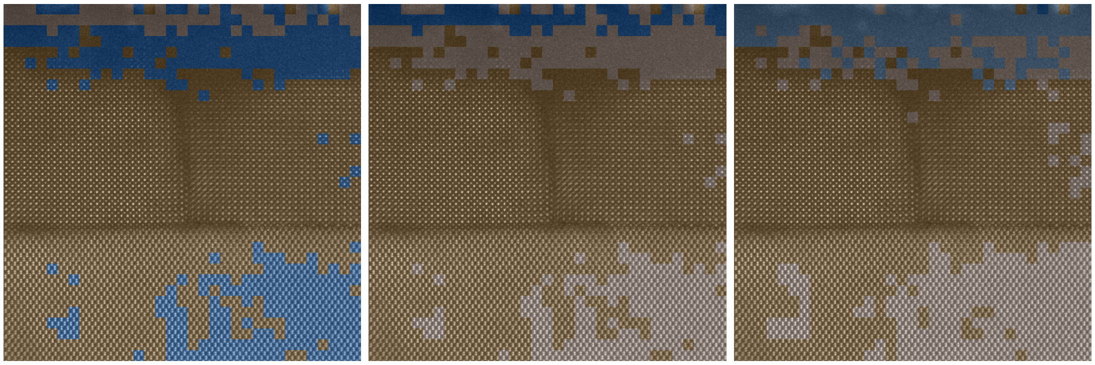

Figure 4. Chipped images colored based on detected Louvain communities when the resnet34 classifier uses (from left to right) (a) no preprocessing, (b) PyTorch normalization, and (c) CLAHE preprocessing. The distance used is scaled to be in the unit interval where 0 is no similarity and 1 is high similarity. This distance is also used as the weight for the edges in the network graphs corresponding to each image above. Note that although (a) and (b) are colored differently, they are still identical since the colors are merely swapped. This is what we observed in Figure 1 using the classifier output. Images (a) and (b) show detection of the topmost interface.

2.4 Spectra Extraction

To incorporate energy dispersive spectroscopy (EDS) data as an additional modality for the pyCHIP classifier, the data must first be extracted from Thermo Scientific Pathfinder X-ray Microanalysis Software as an easily accessible format. Included in the software package is a file conversion program known as SIConvert.exe which converts the data files into. raw and .rpl files. After the data was extracted, a preliminary multimodal analysis was performed. The outcome of this analysis is presented in the figure below

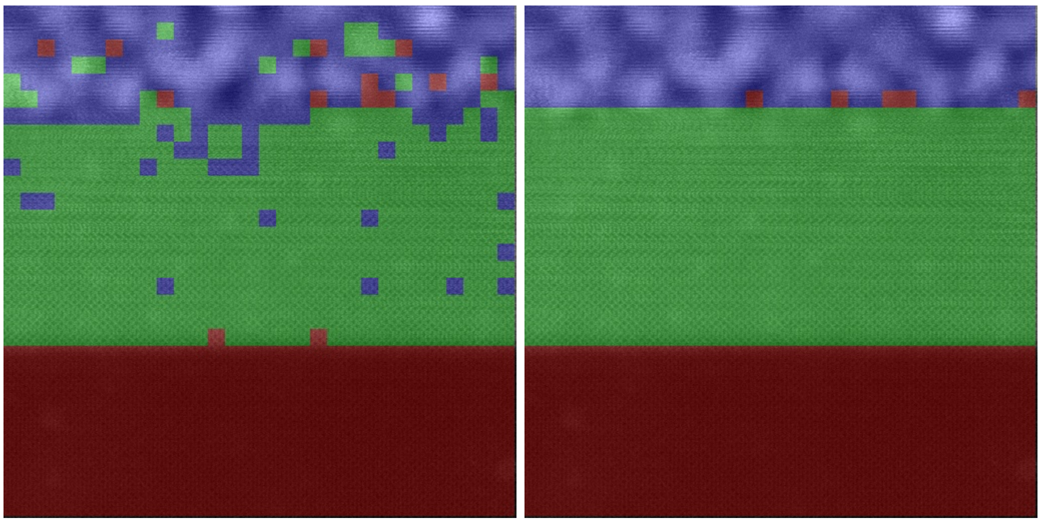

Figure 5. Chipped images colored based on predicted labels when classifier uses (from left to right) (a) unimodal data, (b) multimodal data. Image (a) is the result of training and testing the pyCHIP classifier using only microscopy image data whereas image (b) is the result of using both microscopy image data and EDS data. Notice how the multimodal classifier does a better job than the unimodal classifier at predicting the labels for this chipped image, and thus has a better accuracy.

3.0 Conclusions

While the pyCHIP classifier needs additional improvements to increase its capabilities, further research on segmentation and multimodality will be fruitful in revealing new insights which can aid in this endeavor. Combinations of different image preprocessing techniques, encoders, and clustering algorithms could allow for better segmentation than what was seen in this report. Furthermore, the multimodal approach will need further development and testing to see if it can extrapolate to new data without suffering significant loses in accuracy. With these advancements, the pyCHIP classifier may prove to be useful in high-throughput analyses soon.

4.0 References

Akers, S., Kautz, E., Trevino-Gavito, A., Olszta, M., Matthews, B. E., Wang, L., . . . Spurgeon, S. R. (2021, November 17). Rapid and flexible segmentation of electron microscopy data using few-shot machine learning. *npj Computational Materials*. Retrieved from https://www.nature.com/articles/s41524-021-00652-z

Doty, C., Gallagher, S., Cui, W., Chen, W., Bhushan, S., Oostrom, M., . . . Spurgeon, S. R. (2022, February 15). Design of a graphical user interface for few-shot machine learning classification of electron microscopy data. *Computational Materials Science, 203*. Retrieved from https://www.sciencedirect.com/science/article/pii/S0927025621007874

Kalidindi, S. R., & De Graef, M. (2015). Materials Data Science: Current Status and Fututre Outlook. *Annual Review of Materials Research, 45*, 171-193.

Olszta , M., Hopkins, D., Fiedler, K. R., Oostrom, M., Akers, S., & Spurgeon, S. R. (2022, June 10). An Automated Scanning Transmission Electron Microscope Guided by Sparse Data Analytics. *Microscopy and Microanalysis, 28*(5). Retrieved from https://www.cambridge.org/core/journals/microscopy-and-microanalysis/article/an-automated-scanning-transmission-electron-microscope-guided-by-sparse-data-analytics/2925BA91037B33A7523ABFEEEB39F6DA

Wang, N., Freysoldt, C., Zhang, S., Liebscher, C. H., & Neugebauer, J. (2021, September 1). Segmentation of Static and Dynamic Atomic-Resolution Microscopy Data Sets with Unsupervised Machine Learning Using Local Symmetry Descriptors. *Microscopy and Microanalysis*, 1-11.

Pacific Northwest National Laboratory

902 Battelle Boulevard
P.O. Box 999
Richland, WA 99354

1-888-375-PNNL (7665)

www.pnnl.gov