Machine-learning aided image recognition and tagging

Ronald Lashley

Department of Computer Science, Lincoln University of PA, Lincoln University, PA, 19352

Bo Sun

Department of Computer Science, Lincoln University of PA, Lincoln University, PA, 19352

Dantong Yu

Computer Science Initiative, Brookhaven National Laboratory, Upton, NY 11973

Abstract:

In technology, image recognition software uses various methods to extract information from an image. Seeking to deliver a data analysis pipeline for x-ray synchrotron instruments to assist more ambitious materials discovery experiments, we aimed to build automated analysis pipelines for extracting scientifically meaningful insights from datasets relevant to materials discovery, especially x-ray scattering images. Because users are unfamiliar with the underneath machine learning toolkit, which requires configuring many parameters, developing a graphical interface will greatly improve the usability of machine learning. Previously, the method for image tagging required the use of command line. By introducing a simple button click, allowing users to choose options, run machine learning tools, visually inspect the results, and automatically generate configuration files. Proposing a transformative new paradigm for scientific research, where data analysis tasks (such as pre-process image, extracting features, and tagging image) are massively automated, will thereby liberate scientists to focus on deep scientific questions. This vision can only be realized through a deep integration of machine-learning procedures into all aspects of data interactions. By eliminating the friction inherent in data analysis and pattern recognition, this development will accelerate science discoveries by shorting time to analyze the data with it being automated now. The tag file (.xml) will be stored together with each image. The downstream pipeline relies on this tag information to perform further analysis, such as navigating through all images, searching images with semantic features (tags). All tag information can be used to create an accurate estimation of the distributions of physics property. Users only need to do a few experiment to obtain the optimal solution. From this project, I have gained valuable experience learning and utilizing Python in applications such as Anaconda Navigator, Python IDLE, and QT Designer.

Introduction:

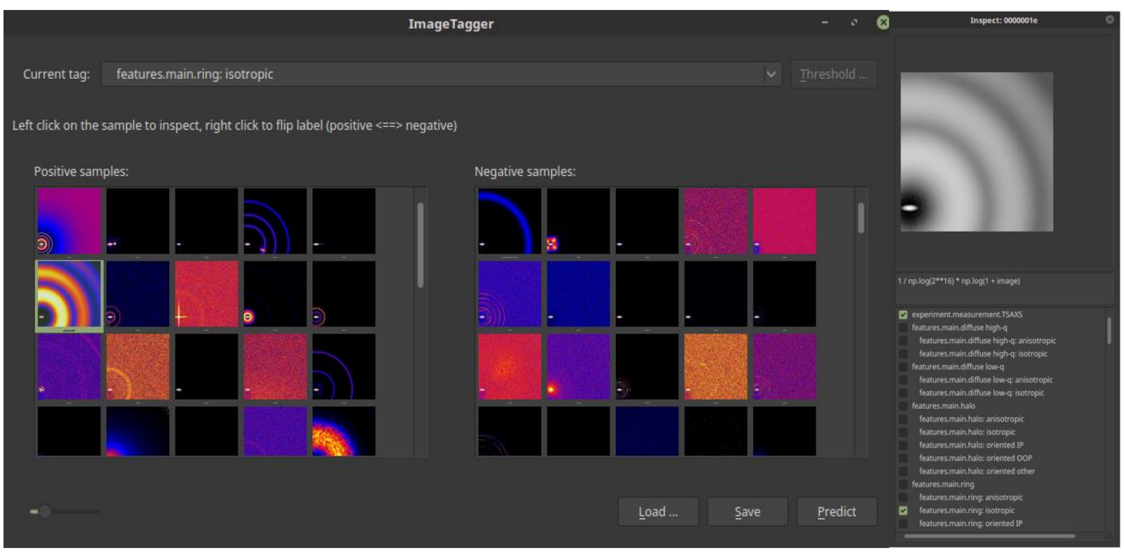
X-ray scattering is a powerful technique for probing the physical structure of materials at the molecular and nanoscale, where strong X-ray beams are shined through a material to learn about its structure at the molecular level. This can be used in a wide variety of applications, from determining protein structure to observing structural changes in materials. Modern x-ray detectors can generate 50,000 to 1,000,000 images/ day, thus it's crucial to automate the workflow as much as possible.

The current workflow in an x-ray scattering experiment consists of an experimental team traveling to a synchrotron beamline, capturing a detailed dataset over several days, and then returning to their home institution with the images for later analysis. The goal is to speed that process up using machine learning and computer vision to automate the process of image analysis. Developing a set of intelligent automated methods, which will be the "brain" of a computer-directed beamline experiment for users to use to be faster and more efficient.

Machine Learning itself is undergoing a shift, with a re-thinking from traditional, naive, neural networks, towards deep learning models where the neural hierarchy is more rational, optimized, and informative. This has already led to clear advances in several fields including computer vision and speech recognition, and we aim to demonstrate similarly transformative gains with respect to scientific image streams. The core idea in deep learning is to design multiple levels of representations corresponding to a hierarchy of features, wherein the high-level concepts and knowledge are derived from the lower layers.

Methods:

The Graphical User Interface has four buttons and a combo box for the user to use. The combo box allows the user to choose which tag they would like to use for the images. The load button allows the user to select a directory of images and scan the record to determine which images have the current tag selected. With this being done in the background, up front, the images with the current tag are displayed in the left panel and the images that do not have that attribute are displayed in the right panel. The predict button triggers the automated analysis tasks (pre-process image, extracting features, and tagging images) allowing the scientist to focus on deep scientific questions. The save button allows the user to save a tag file as a .xml that will be stored with each image. The downstream pipeline relies on this tag information to perform. This only needs the user to do a few experiments to obtain optimal solution.



1

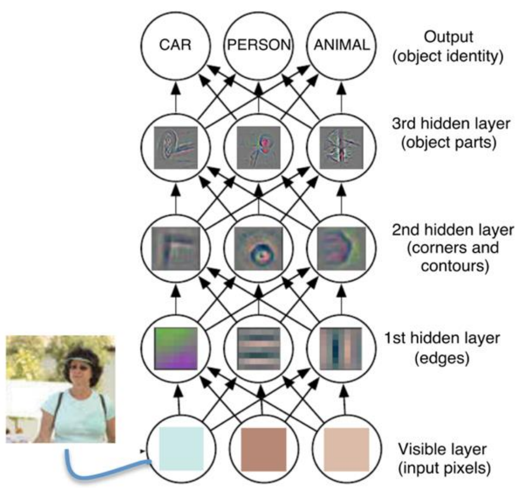
The Image Tagger graphical user interface, figure 1, allows the user to load a dataset/ set of images. When a directory is chosen, the thumbnails of the images are displayed with the false color. The false color enhances the image so that the user can visualize the features to better identify them. For a set of images that were previously tagged, when a tag is selected from the combo box, the images are sorted so that the images with the tag feature are placed in the Positive sample window and the ones without go into the Negative sample window. For a new set of data/ images, the predict button triggers the automated analysis tasks. This will predict the tags and then sort them in the positive and negative sample windows. With there being chance of error, there is an Inspect window that is open along with the Image Tagger. The Inspect Window consists of the real image display which is a grayscale image, a text box to insert a line of python code to change / enhance the image, and a list of all tags that the user can use to check or uncheck. The user can click on an image in either sample window to display the real image and use the Inspect window to check the tags that they know the image has. Another way for the user to make corrections are by right clicking the thumbnail image. By right clicking the thumbnail image, the image goes from one window to the next. This action adds the tag to the image or removes the tag from the image.

For machine-learning, there are two types of datasets that are used. The first is real dataset, which is collected by shining powerful x-rays through a particular material and the attribute is labeled by material experts. The second type dataset is synthetic scattering dataset, where the data is generated by simulation software. The simulation software is able to synthetic scattered images based on physics laws. The real dataset consists of 2832 gray-scale x-ray scattering images, contains images from 13 x-ray scattering measurements runs. These measurement runs are a set of related x-ray scattering images collected for closely related material samples. All of the images have been labeled with 104 binary attributes (tags)[2] by a domain expert. These attributes represent a diverse set of characteristics ranging from the type of measurement, to appearance based scattering features, to chemical composition and physical properties of the materials. Tags can include a diverse selection of images, which makes classification of x-ray scattering images difficult.

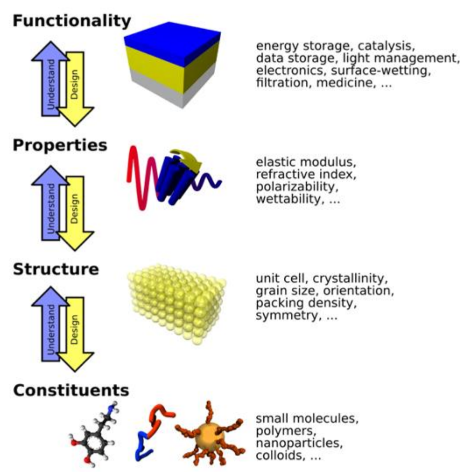
Attribute	#	Attribute	#
Thin film [G5]	1646	Silicon [G7]	130
Specular rod [G3]	1597	GTSAXS [G1]	127
Beam off image[G2]	1591	MWCNT [G7]	125
Photonics CCD[G2]	1591	Nanoporous [G5]	125
Ordered [G5]	1462	Theta sweep [G1]	109
GIWAXS [G1]	1439	PDMS [G7]	107
MarCCD [G2]	1241	Saturation artifacts [G3]	97
Horizon[G4]	1171	Peaks: Line z [G4]	90
Linear beamstop [G2]	1156	Circular beamstop [G2]	85
Peaks: Isolated[G4]	1099	Peaks: Line xy [G4]	79
GISAXS[G1]	870	Diffuse low-q: Anisotropic [G4]	78
Ring: Oriented z [G4]	856	Many rings [G4]	78
Polymer [G6]	821	Diffuse low-q: Oriented z [G4]	76
Halo: Isotropic [G4]	791	Misaligned [G3]	76
Ring: Isotropic [G4]	604	Beam streaking [G3]	70
Ring: Textured [G4]	528	Diffuse low-q: Oriented xy [G4]	69
Higher orders: 2 to 3 [G4]	513	Blocked [G3]	62
P3HT [G7]	505	Diffuse specular rod [G4]	62
Ring: Oriented xy [G4]	491	Smeared horizon [G4]	55
SiO2 [G7]	467	Symmetry ring: 4-fold [G4]	55
Vertical streaks [G4]	434	Higher orders: 10 to 20 [G4]	53
Single crystal [G5]	430	Ring doubling [G4]	53
Block-copolymer [G6]	416	Halo: Anisotropic [G4]	46
Peaks: Many/field [G4]	396	Powder [G5]	44
Grating [G5]	375	Specular rod peaks [G4]	41
PCBM [G7]	369	AgBH [G7]	40
Diffuse high-q: Isotropic [G4]	357	Ring: Oriented other [G4]	33
Higher orders: 4 to 6 [G4]	351	Peaks: Line [G4]	23
Weak scattering [G3]	318	Diffuse high-q: Oriented z [G4]	20
Rubrene [G7]	266	Bad beam shape [G3]	19
TSAXS [G1]	264	LaB6 [G7]	16
Higher orders: 7 to 10 [G4]	260	Phi sweep [G1]	16
2D detector obstruction [G3]	224	Peak doubling [G4]	15
Bragg rods [G4]	211	Halo: Oriented xy [G4]	14
Ring: Anisotropic [G4]	205	Polycrystalline [G5]	14
Peaks: Along ring [G4]	201	Diffuse high-q: Oriented xy [G4]	11
Amorphous [G5]	197	Direct [G3]	11
Saturation [G2]	193	Object obstruction [G3]	9
PS-PMMA [G7]	190	Peaks: Line other [G4]	9
Composite [G5]	179	Waveguide streaks [G4]	8
Diffuse low-q: Isotropic [G4]	170	Higher orders: 20 or more [G4]	4
Yoneda [G4]	167	Substrate streaks/Kikuchi [G4]	4
Strong scattering [G3]	159	Diffuse low-q: Oriented other [G4]	3
TWAXS [G1]	152	Halo: Spotted [G4]	3
Halo: Oriented z [G4]	148	Diffuse low-q: Spotted [G4]	2
High background [G4]	142	Diffuse high-q: Spotted [G4]	1
Asymmetric (left/right) [G2]	138	Empty cell [G3]	1
Ring: Spotted [G4]	136	Parasitic slit scattering [G3]	1
Superlattice [G6]	136	Point detector obstruction [G3]	1

. [2]

Synchrotron images analyzed using a combination of existing domain and image-analysis techniques, as well as new algorithms. (Supervised/Unsupervised) Cluster and tag the data with physically-meaningful attributes. Attributes/features are used to extract higher-order trends, and to extract scientifically-relevant insights. For example, this procedure could be mapped to a four-layer convolution neural network for trend analysis. Figures 3 and 4 are comparisons of hierarchies underlying physical systems and deep learning models. Constituents organize into well-defined structures, which give rise to emergent properties, which in turn dictate functional response. By developing a machine-learning hierarchy closely aligned with this physically relevant hierarchy, we will enable meaningful insights to be automatically extracted from scientific data at multiple levels. Similarly, deep learning recognizes raw images represented as a collection of pixel values to an object identify by breaking the complicated mapping into a series of simple mappings recursively, which creates a hierarchy. The input pixels are fed to the bottom visible layer, then a series of hidden layers abstract complex features from the lower layer. Both of them demonstrate similarity in feature extractions.



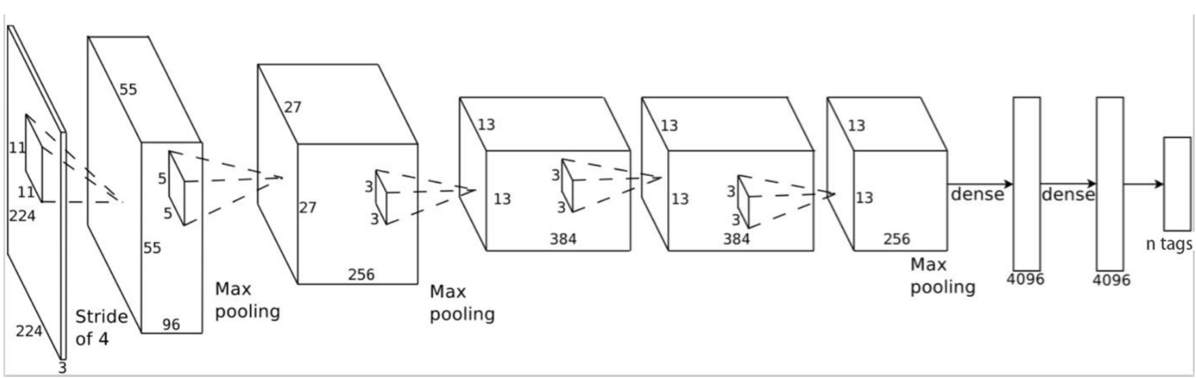
3



4

Deep learning can extract meaningful insights and detect patterns from massive amount of data; well-suited to image-like datasets. Deep learning trains multiple levels of features/ representations to extract meaning from data. The exploration of machine-learning hierarchies tuned to extract physics layers and meaning from scientific datasets. Deep learning has achieved huge success for image classification in recent years. This breakthrough is mainly because of huge amount of data available and good network architecture. Alex-Net is the first Convolutional

Neural Network that has successfully applied in this area. Neural Network can be used to learn features for image classification instead of using hand-crafted image features. The basic units in Convolutional Neural Network are the convolutional layer, subsampling/ pooling layer, and the activation layer. At the convolutional layer, if you just apply fully connected network, it requires lots of parameters. Instead, convolutional neural network is applied, which allow parameter sharing to reduce the number of parameters and discovering local patterns. Subsampling/ pooling layer allows local patterns to pooled in a large region. The activation layer is a non-linear Activation allowing the neural network to learn nonlinear functions. Traditional activation Layer utilize sigmoid function. However, the problem of sigmoid is that the gradient will decrease. Besides, sigmoid is easy to saturate, when it’s saturated, the gradient is close to 0. If the network has multiple sigmoid layers, the gradient at early layer will be every small, which make the parameters of early layers hard to optimize. Instead, the most widely used activation function is Rectified-Linear Units (ReLU). If the value is below 0, the output of ReLU is 0, otherwise, the output is the same as input. Figure 5, is the CNN Network architecture overview. We use Alex-Net based architecture, which contains 5 layer convolutional layers. The last layer was changed from softmax layer to sigmoid layer, since one image could have multiple tags



5

Conclusion:

In this work, we addressed the development of a graphical user interface that is simple for an experimenter that is unfamiliar with the underneath machine learning toolkit which requires configuring parameters. Previously being required to use command line clients to tag image files and having to manually inspect each image, and use editor to inspect the corresponding tag file, users can now use a GUI to choose options, run machine learning tools, visually inspect the results, and automatically generate configuration files.

Machine-learning methods liberates scientists to work on science, enables computer-controlled ‘intelligent’ exploration of materials questions, and accelerate scientific discoveries. Deep-learning is a crucial tool, allowing the computer to extract physically-relevant meaning from abstract datasets. This elimination of friction inherited in data analysis and pattern recognition, this development will accelerate science discoveries. Machine-learning is a critical component of automated materials discovery and will continue to improve and grow therefore requiring the GUI to be flexible. The GUI being flexible allows changes to be made without having to make a new one.

The next step is to create more images to train the Convolution Neural Network. The more data/ images used to train the CNN will reduce and eliminate the overfitting. Also, we will test the network trained on synthetic dataset on real X-ray images to automate the current workflow as much as possible.

Acknowledgements:

This project was supported in part by the National Science Foundation, Louis Stokes Alliances for Minority Participation (LSAMP) at Lincoln University of Pennsylvania under the LSAMP Internship Program at Brookhaven National Laboratory. I wish to thank my host, Dr. Dantong Yu, for his professionalism and generosity during the NSF program. I would also like the thank my Visiting Faculty Professor and Mentor, Dr. Bo Sun, for her assistance, guidance, and for selecting me as the student to join her team. Further, I would express my gratitude to the Office of Science Education, and all who continue to so willingly assist interns in that branch. I very much appreciate the efforts of the National Science Foundation, LSAMP at Lincoln University of Pennsylvania with regard to their support. Finally, I wish to acknowledge the hospitality and kindness of Brookhaven National Laboratory and the Department of Energy.