Title
Convolutional Neural Networks at the Interface of Physical and Digital Data

ISBN
9781509032846

Authors
Ushizima, Daniela
Yang, Chao
Venkatakrishnan, Singanallur
et al.

Publication Date
2016-10-01

DOI
10.1109/aipr.2016.8010606

Peer reviewed

Convolutional Neural Networks at the Interface of Physical and Digital Data

Daniela Ushizima*‡, Chao Yang*, Singanallur Venkatakrishnan†, Flavio Araujo*‡, Romuere Silva*‡,
Haoran Tang§, Joao Vitor Mascarenhas*, Alex Hexemer†, Dilworth Parkinson†, James Sethian*§
*Computational Research Division, Lawrence Berkeley National Laboratory
†Advanced Light Source, Lawrence Berkeley National Laboratory
‡Berkeley Institute for Data Science, University of California Berkeley
§Mathematics Department, University of California Berkeley

***Abstract*—Electron and X-ray interactions with matter can be recorded as digital images, which are signal acquisition mechanisms often used to investigate materials microstructure. Recently, the ability to quickly acquire large datasets at high resolution has created new challenges in areas that rely upon image-based information. The proposed analysis schemes employ Convolutional Neural Networks as the core algorithm in the reconnaissance of expected events from data gathered in two regimes: experimentally and by simulation. At the interface of physical and digital datasets, we propose classification schemes that exploit complex geometrical structure from scientific images through different machine learning packages, such as MatConvNet and TensorFlow. Our results show correct classification rates over 90% considering thousands of samples from four image modalities: cryo-electron microscopy, X-ray diffraction, X-ray scattering and X-ray microtomography. Our main contributions are: (a) developing algorithms designed for data that stem from physical experiments; (b) building new software to constrain parameter space, particularly given new hardware; and (c) testing different CNN models for classification of scientific images[1].**

I. Introduction

Recent estimates for daily data production are around 2.5 quintillion (10^{18}) bytes, with approximately 90% of the world's data having been generated over the past two years [1]. For example, instrument upgrades with new accelerator and signal detector technologies at DOE national laboratories' experimental facilities such as synchrotrons and neutron sources [2] have led to an unprecedented increase in the quantity and complexity of output data.

This increase poses a significant challenge to experimental scientists who need to analyze and interpret these data to gain scientific insights and understand of a variety of phenomena. Over the last few decades, scientists have accumulated a great deal of experience in data analysis, however this experience has yet to be turned into analytical software tools. Machine learning through artificial neural networks is one way to encode the experience and knowledge scientists have developed into a flexible machinery that can be used to analyze new data and observations. This type of inferential engine can be constantly updated as new experience or knowledge is gained.

Some of the experimental data generated at DOE facilities are in the form of images, which are particularly well-suited for convolutional neural networks (CNN). CNN goes beyond conventional computer vision approaches [3, 4, 5] to address the semantic gap between signal acquisition and image interpretation [6]. The major challenge in using CNN is to come up with appropriate labels and descriptors for existing data that have been processed, and to construct CNNs that learn from these labeled datasets to make intelligent predictions and characterization of new data to be collected in future experiments.

In this paper, we describe several different types of image analysis problems arising from DOE experimental facilities that can benefit from the use of CNN. Table I summarizes the use-cases reported in this paper, which consists of cryo-electron microscopy, X-ray diffraction, X-ray scattering and X-ray microtomography. We will describe the sources and characteristics of the image data, which are quite different from handwriting recognition or web images, and discuss how these data are labeled. We present preliminary results on how CNNs are constructed and trained for these datasets, which software packages we have used, and how the CNNs constructed with these tools perform for the type of analysis required by experimentalists.

Ideally, deep learning capabilities could be built into the instruments so that data acquisition and analysis can be combined in a seamless fashion. In addition to using conventional CPUs or GPUs, which typically consume a tremendous amount of power, we have explored the use of light weight processors that are power efficient. One way to achieve power efficiency is to lower the precision requirement of the input data and CNN parameters. We describe a quantization algorithm to achieve this goal. We also discuss work on using the IBM TrueNorth chip, which embodies a neuromorphic architecture that is drastically different from the traditional Von Neuman architecture used in existing CPUs and GPUs.

The use-cases reported in this paper focus on scientific images coming from instruments based on cryo-electron microscopy and different X-ray-based imaging techniques, as described in Section II, and also summarized in Table I. Section III discusses the algorithms and software packages that support the implementation of pattern recognition processes, and Section IV presents details about parameters and accuracy results associated to each use-case. Section VI explains the

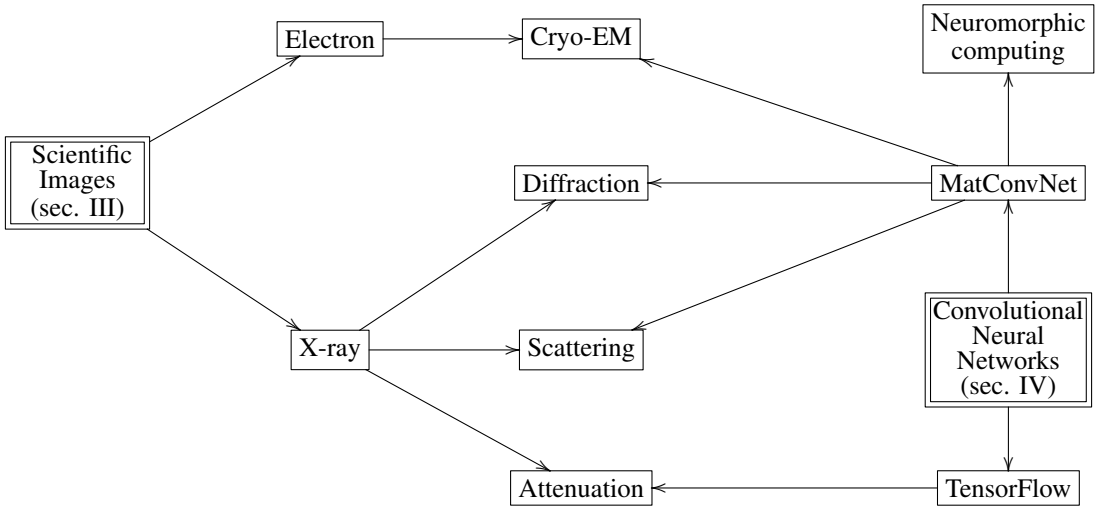


Fig. 1: CNN for Experimental and Observational Data

TABLE I: Scientific data under scrutiny with CNN: specifications and methods

Specimen	Scale range	Image modality	Imaging mechanism	Data analysis for quantitative microscopy
Transcription factor (TFIID)	10 – 100 nm	cryo-EM	electron scattering	Inspection of molecular structure with 2D orientation classification using MatConvNet. Sec. II-A. Fig. 2.
Photosystem II	1-10 nm	crystallography	X-ray diffraction	Identify diffraction images that contain Bragg peaks using MatConvNet. Sec. II-B1. Fig. 8.
Thin films	1.64 nm	GISAXS	X-ray scattering	Classification of crystal lattice structure using MatConvNet. Sec. II-B2. Fig. 5.
Fiber beds	0.65 to 1.3	microCT	X-ray attenuation contrast	Detection of fiber profiles from 3D cross-sections using TensorFlow. Sec. II-B3. Fig. 6.

main contributions of our investigation, which includes new algorithms to leverage and/or create data based on Physics, building new software to constrain parameter space, particularly given new hardware, and testing different CNN models for scientific images.

II. Recording and simulating scientific data

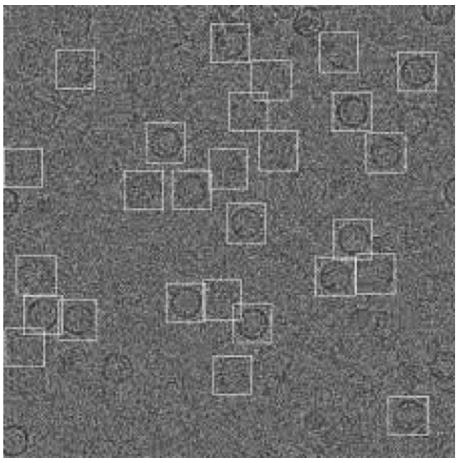
While expected growth in data size and rates vary among science areas, recent reports [2] suggest that several imaging facilities will soon be struggling with 1 to 50 petabytes worth of data generated per year. This much data poses several challenges: (a) inadequate or insufficient meta-data describing experimental records; (b) the impracticality of manually curation of massive datasets; and (c) the lack of tools adapted to new data acquisition modes.

In order to address scalable solutions to scientific data that goes beyond deploying one-off tools, we have explored deep learning to tackle different scientific images, as described in following sections. The imaging techniques are quite different in their electromagnetic wave interaction with the objects and the range of applications, but all require nontrivial mathematical methods to recover and understand the resulting images composition and microstructure.

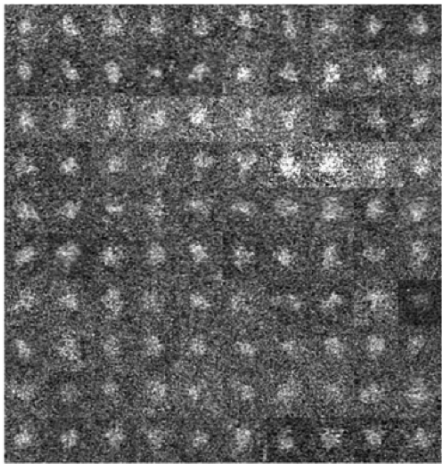
A. Case I: Cryo-EM

Cryo-electron microscopy (cryo-EM) is a technique for elucidating three-dimensional structures of macromolecules and molecular complexes. In a cryo-EM experiment, a large number of isolated biomolecules in the same conformational state are placed in a thin layer of vitreous ice. The prepared specimen is then placed into an electron microscope that produces a number of two-dimensional (2D) projected views of the molecules through weak phase electron scattering. Because molecules take random orientations when prepared in a sample, the relative orientation of the 2D views are unknown at the time of imaging. They must be determined computationally together with the 3D structure of the molecule. Because this technique does not require molecules to be prepared in crystallized form, it is also referred to as single-particle method.

Figure 2 (a) shows a typical micrograph collected from a cryo-EM experiment. The micrograph contains a number 2D views of the molecular specimen. Because the electron dose of the microscope has to be sufficiently low in order to prevent the samples from being damaged by radiation, the signal-to-noise ratio (SNR) in the cryo-EM image is extremely low. As a result, one can barely distinguish the particle from the



(a) A cryo-EM micrograph containing projected views of macromolecules.



(b) 2D projected views of a TFIID molecule.

Fig. 2: Examples of cryo-EM images.

background, as seen in Figure 2.

One of the first tasks in processing the cryo-EM images consists in isolating the particles from the background in each micrograph, a step known as particle picking or boxing. Although a number of techniques have been developed in the last thirty years to perform this task in an automated fashion, none of the existing algorithms is completely satisfactory or reliable. Human intervention is often required in order to discard sub-images that do not contain particles, a painstaking process.

In addition to particle picking, a CNN can be used to perform a more complex step which involves classifying boxed particle images into distinct projection orientation classes. By averaging images within the same class, we will show that class averages with higher SNRs can be used to construct an initial low resolution 3D structure of the macromolecule. Such a low resolution model can subsequently be refined in a nonlinear iterative optimization procedure [7].

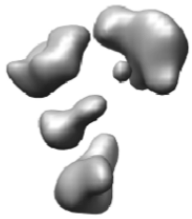
Figure 3b shows nine distinct projected views of the macromolecule transcription factor IID (TFIID). These views are generated by computationally projecting a previously reconstructed 3D density map of TFIID, as shown in Figure 3a along nine uniformly distributed projection directions, defined by the set of Euler angles shown in Table II. In practice, the 2D images contain much more noise. Figure 2(b) shows some of the experimental images produced by a particle picking program. These images also contain a background that corresponds to the projected density of substrate (ice).

Multivariate statistical analysis tools such as principle component analysis (PCA) and K-means are popular algorithms for solving the 2D classification problem. However, because it is difficult to quantify the image formation model, these tools lack sufficient accuracy to account for variations of the 2D images due to differences in imaging conditions. In many cases, 2D images placed in the same class need to be visually examined and outliers that do not belong to that class must be removed.

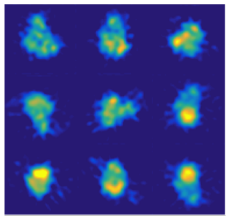
TABLE II: Euler angles used to generate projected views of a previously reconstructed TFIID molecule.

$\Psi \downarrow_i$	$\Theta \downarrow_i$	$\phi \downarrow_i$
0.00	0.0000	0.0000
0.00	45.000	0.0000
0.00	45.000	89.975
0.00	45.000	179.95
0.00	45.000	269.92
0.00	90.000	0.0000
0.00	90.000	59.983
0.00	90.000	119.97
0.00	90.000	178.95

In this paper, we test the ability of a CNN to provide a mechanism to encode both multivariate statistical analysis based classification techniques and human vision based classification skills in a single machinery that can be constantly updated by labeled image data coming from simulation and experiments.

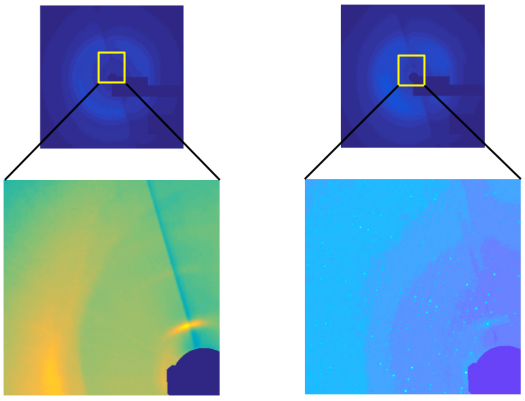


(a) A previously reconstructed 3D electron density map of a TFIID molecule.



(b) Nine distinct 2D projection views of the TFIID molecule.

Fig. 3: Simulated cryo-EM projection images from a previously reconstructed 3D model of the TFIID molecule.



(a) Hit (b) Miss

Fig. 4: Two examples of X-ray diffraction images. the image on the left does not contain any visible Bragg spots probably due to X-ray beam missing the sample. The one on the right contains Bragg spot which can be seen in the zoom-in image.

B. Imaging Techniques based on X-ray

Synchrotron radiation relies on a charge moving at relativistic speeds, and following a curved trajectory [8]. The x-ray experimental data used in this article was acquired at the Advanced Light Source, which is a Department of Energy-funded facility that provides users from around the world access to the brightest beams of soft x-rays, together with hard x-ray and infrared light, for scientific research and technology development in a wide range of disciplines [9].

1) Case II: X-ray diffraction and Bragg peaks: X-ray crystallography is a technique for determinating the 3D structure of biomolecules when these molecules can be crystallized. The diffraction of X-ray beams off a crystal sample produces an image with many bright spots that are often called Bragg spots or peaks. These spots correspond to the intersection of a so called Ewald sphere and the reciprocal space lattice. Identifying the positions of these Bragg spots enables characterization of the crystal type and lattice parameters. The quality of the diffraction image depends on a number of experimental factors and how well the sample is prepared. In many cases, diffraction images lack clear Bragg spots, therefore these images are often discarded. Other scenarios in which the image is unusable because bright spots are artifacts instead of real Bragg spots.

Although domain scientists can recognize the presence of Bragg peaks in most cases by human vision, going through all images manually has been a serious constraint. Current automated tools available in the community provide limited reliability. Training a CNN with labeled images may be an alternative to several of the cases in which we have enough data to separate images with clear Bragg spots from others that lack good quality spots.

2) Case III: X-ray scattering and GISAXS: Scattering techniques enable probing micro and nano-structures with a high degree of statistical relevance. The most prominent techniques are small angle X-ray and neutron scattering (SAXS and SANS) for the detection of mesoscale structures as well as wide angle X-ray and neutron scattering (WAXS and WANS) for the investigation of the structure down to the molecular scale. Prominent examples are thin films of conductive polymers, organic electronics, or thin films of nano-particles, which receive high interest for energy relevant materials such as organic photovoltaics. Morphological characterization of thin films is challenging, and substantial advancements have been accomplished through research efforts on new experimental geometry setups for scattering. Unfortunately, these have not been enough to provide the required data reduction and analysis that is fundamental for giving domain scientist access to use advanced grazing incidence techniques and getting the most relevant information out of their experiments. In particular in-situ analysis is far from possible at the given time.

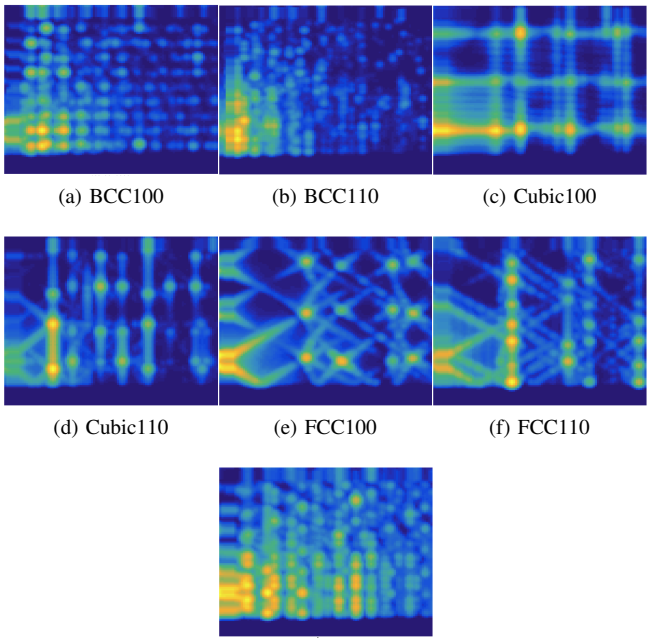
As a first step towards using machine learning for GISAXS analysis, we address the problem of identifying one useful property of certain types of samples - the type of crystal structure corresponding to the measured scattering pattern.

We propose a supervised classification approach to identify the crystal lattice types from a Grazing Incidence Small Angle X-ray Scattering (GISAXS) image. GISAXS, an important reciprocal-space imaging modality, provides statistical information about a sample in 3-D. GISAXS is widely used for studying thin films that play a vital role as building blocks for the next generation of renewable energy technology. One challenge in GISAXS imaging is to accurately infer the crystal lattice corresponding to the sample from a single 2-D diffraction/scattering pattern.

As a first step towards understanding crystal configurations, we used the HiPGISAXS [10] simulation package to generate a large collection of sample images from each class of possible crystal structures and test the algorithm performance under multiple simulated test images. Inspired by the recent successes of deep-learning approaches for natural image classification, we tested CNN as the method to carry out the classification of large physically realistic synthetic datasets.

Our peak-position detection exploration from image datasets retrieve information about the lattice and orientation of the material. A typical image from the ALS beamline 7.3.3 presents shapes used as input to derive the underlying structure of the sample. The location, shapes and position of these structures, while yielding useful information, can also be used as an input to simulation tools. Shapes vary among rings, arcs, peaks and lines. In case of time resolved or multi-energy data, the features become three or more dimensional, e.g., peaks become rods and lines become planes.

3) Case IV: X-ray attenuation contrast and microtomography: Another example of high-throughput data collection is hard X-ray microtomograph (microCT) imagery, acquired for the structural investigation of dense materials, such as ceramic matrix composites (CMC). These hierarchical materials consist of many individual strands, bundled within a matrix to achieve high-strength mechanical properties and durability. In order to test the operating conditions for which these materials are



(a) BCC100 (b) BCC110 (c) Cubic100

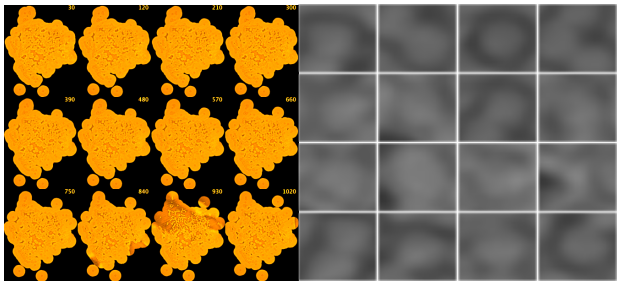
(d) Cubic110 (e) FCC100 (f) FCC110

(g) HCP

Fig. 5: GISAXS scattering patterns corresponding to seven different crystal classes generated using a simulation software. We use such simulated data to design a classifier for the seven crystal types.

designed, they are heated to nearly 2,000 degrees and then stressed, causing several deformations [11, 12]. Currently, several materials scientists collect a series of 3D images as a function of time, using a hot cell to vary temperature and tensile forces. After performing dynamical experiments (3D+time), the 3D images are examined manually, slice by slice (2D), looking for the individual defects that the material may form, which one can correlate with decreased performance.

This process is prone to error and very time-consuming. Alternatively, we would like to automate this task by designing "virtual inspectors", i.e., algorithms that detect mechanical deformations from microCT automatically, and then point out which of the experimental instances (image stacks) present the structures of interest, e.g., matrix cracks. Being able to detect these properties in real time will add an entirely new level of experimental capability. Tying this capability to the control systems at the beamlines may allow the experiment to be steered by the machine in response to specific structures present in the sample. For example, when a feature of interest is identified, the imaging process may be temporarily enhanced, so that a magnified image is collected from the region of interest, before resuming the process. Currently, a major challenge is the inability of adjusting experimental parameters fast enough for optimal data collection, therefore the users cannot manage the acquisition of detailed features of interest.



(a) Stack cross sections (500μm) (b) Fiber profiles

Fig. 6: MicroCT of ceramic matrix composite cross-sections .

Figure 6 illustrates CMC sample cross-sections, containing sample about the size of a long-rice grain, with 1mm diameter and 55mm length, reinforced with 500 Hi-Nicalon fibers of approximately 10μm diameter, bounded together by a chemical vapor infiltrated SiC matrix. Each fiber is coated with a boron nitride (BN) layer through chemical deposition prior to chemical vapor infiltration of the SiC matrix. The BN layer acts as a weak interface for the matrix cracks to deflect and to ease fiber sliding. Due to the BN layer around the fibers, which has a lower X-ray absorption coefficient, fiber profiles appear as dark rings. Figure 6(b) emphasizes several fiber profiles, commonly used as a guiding pattern to detect other fiber cross-sections. A major demand is the ability to perform pattern ranking mechanisms, which can steer algorithms such as template matching [12, 13], and improve data management by beamline scientists.

Among these labeled images, 200,000 contain fibers and 100,000 images present areas with no fibers. All the labeled samples passed through a triage consisting of two steps: automated segmentation methods based on traditional computer vision [14, 11, 12] and visual inspection by more than three domain scientists. To the best of our knowledge, all the images contain 100% accurate labels, which are used to determine the success rate of the CNNs.

III. Convolutional Neural Networks

Synapses enable nerve cells to connect to thousands of other neurons, combine signals, and push the integrated information forward. Early research by Hubel and Wiesel [15] on the cats visual cortex showed that neurons propagate information through complex cell organizations called receptive fields [16]. These fields work similarly to filters for the stimuli coming from previous retinal layers, and allow information from 125 million photoreceptors to be successfully processed and pushed forward to 1 million ganglion cells along the visual processing path. These biological filters can be framed into algorithmic motifs that exploit the strong spatially local correlation present in digital images for the purpose of learning tasks.

Artificial neural networks (NN) provide approximation methods to better understand many complex real-world sensor data through learning of real-valued, discrete-valued, and/or

vector-valued functions from data [17]. In 1982, NN emerged as a computational procedure that combine simple components, similar to neurons, capable of representing physical meaning of content-addressable memory [18].

As an example of NN algorithms, convolutional neural networks have been successfully used in pattern recognition because CNN can learn a hierarchy of image-based features by building high-level patterns from primitive ones, thereby automating the process of feature construction, much different from traditional descriptor design. The ability to auto-tune kernels parameters through back-propagating errors is one of the reasons CNN turned into the most popular algorithm for computer vision applications [19].

Convolution is an operation that combines two functions, by providing the summation of pointwise products of function values, subject to traversal [4, 20]. The idea of convolution permeates CNN definitions, in which signals/images pass through multiple layers of non-linear operations to produce the pattern classification result. However the operation performed on image inputs with CNNs is not strictly a convolution, but rather its counterpart called cross-correlation [4, 21, 22]. Typically, convolutional kernels are real-value matrices smaller than the input image, responsible for transforming images into activation maps.

Four main types of layers are typically used to build CNNs: convolutional, ReLu (Rectified Linear Unit), pooling and fully connected. This layered architecture relies on well-established calculus, the chain-rule, to adjust layer weights using error minimization, through optimization techniques such as stochastic gradient descent.

A number of tools for constructing CNN are available. We will describe some of these tools that we have used to analyze the data in Section II.

A. Matconvnet

MATLAB has several implementations of CNNs included in the MatConvNet toolboxes used in computer vision and pattern recognition tasks. It is simple to learn, efficient, and permits testing state-of-the-art CNNs quickly. MatConvNet can be used to train models, typically by using a form of stochastic gradient descent and back-propagation. Several examples of CNNs can be found in the public repository[23], which enable image classification and encoding [24] through optimized implementations in MATLAB code. We deployed such codes leveraging architectures previously designed for known datasets, such as MNIST, CIFAR and ImageNet, but here refactored to recognize patterns from scientific images from Section II.

Independently of the toolbox, the architecture of CNNs depends on a large number of data-driven parameters, such as weights derived during a broad search over the hyperparameter space. The memory footprint to compute and store CNN models motivated our team to investigate methods to adapt CNN to energy efficient devices. Among several CNN designs, this paper also reports on data reduction schemes and net weight representations in order to accurately classify scientific data from simulations, aiming to port code to neuromorphic chips.

B. Neuromophic computation

Merolla et al. [25] reported promising results on weight binarization and other non-linear distortions as part of schemes to port neural networks to neuromorphic chips. More recently, Esser et al. [26] discussed mechanisms to model CNNs by going beyond contemporary networks, which use high precision (32-bit and higher) neurons and synapses to support gradient learning based on back-propagation. Instead, neuromorphic designs explore one-bit spikes for event-based computation and communication, then using low-precision synapses and colocating memory with computation – this new conception enables deployment of hardware that consumes a fraction of the energy otherwise necessary. Recent results [26] confirm that arrays of IBM TrueNorth chips are capable of 1,200 to 2,600 frames/s and using between 25 and 275 mW (effectively more than 6,000 frames/s per Watt).

With a unique energy-efficiency chip architecture based on spiking neurons, neuromorphic computation using the IBM TrueNorth [27] has enabled neural network algorithms to represent connections with low precision and guarantee scalable communication pathways.

The IBM TrueNorth chip contains 4,096 cores, 256 axons, 256^2 synapse cross-bars and 256 spiking neurons, with average neuron spiking rate at 8.1 Hz. While the algorithmic design occurs within the Matlab environment, an external workflow called Eedn couples the energy efficiency of IBM TrueNorth with the versatility of deep learning. Eedn stands for energy-efficient deep neuromorphic networks [26], and consists of a powerful set of codes for speeding up the process of creating CNNs adapted to this energy-efficient chip. The job of a TrueNorth programmer is to translate a desired computation into a specified network of neurosynaptic cores, mapping inputs into outputs through corelets.

C. TensorFlow

TensorFlow is an open source software library for expressing machine learning algorithms. Nodes in the graph represent mathematical operations, while the graph edges represent the multidimensional data arrays (tensors) intercommunication. The flexible architecture allows you to deploy computation to a wide variety of heterogeneous systems, including large scale systems based on one or more CPUs or GPUs with a single API. TensorFlow was originally developed by researchers and engineers working on the Google Brain Team within Google's Machine Intelligence research organization for the purposes of designing inferential mechanisms based on deep neural networks, and applicable to several domains, such as speech recognition, computer vision, robotics, information retrieval, natural language processing, geographic information extraction, and computational drug discovery [28, 29].

Our paper reports on preliminary results using TensorFlow to create CNN-based algorithms using the AlexNet [30] architecture. Here, we designed a CNN with two convolutional

TABLE III: Case I: The architecture of the CNN used for CryoEM image classification

Layer No.	Layer Type	Output Size	Filter Size
1	Input	64 x 64 x 2000	—
2	Convolutional	18 x 18 x 100	11 x 11
3	Max-pooling	6 x 6 x 100	3 x 3
4	ReLU	6 x 6 x 100	—
5	Convolutional	1 x 1 x 84	6 x 6
6	Fully connected	84 x 1	—

layers, two pooling layers and two fully connected layers. The two convolutional layers contain 64 filters with size 5x5, working with strides of size 1 and no zero padding. Two layers of max pooling immediately follow the first two convolutional layers, with size 2x2 and and stride 2. The last two layers are fully-connected with 382 and 192 neurons, respectively. The last fully-connected outputs a feature vector, which is ultimately used as signature to retrieve images by similarity, as illustrated in Figure 10.

IV. Results

This section follows the methodology proposed in Figure 1, organizing the four science domains into subsections, similarly to Sec. II, including descriptions of the CNN adopted for each one of these science domains.

A. Case I: Cryo-EM image classification

To test how well a CNN performs in tasks involving distinction among different 2D projection views of a macromolecule, we first generated a set of 2,000 training images by projecting the 3D density map of a previously reconstructed TFIID model along different projection directions defined by Euler angles $(\phi_i, \theta_i, \psi_i)$. Each Euler angle is generated by randomly drawing from 84 uniformly distributed projection directions (with 15 degrees separation) defined by $(\hat{\phi}_j, \hat{\theta}_j, \hat{\psi}_{\theta_j})$ and adding a small angular perturbation, i.e.,

$$\begin{aligned} \phi_i &= \hat{\phi}_{j_i} + \Delta\phi_i, \\ \theta_i &= \hat{\theta}_{j_i} + \Delta\theta_i, \\ \psi_i &= \hat{\psi}_{j_i} + \Delta\psi_i, \end{aligned}$$

where $j_i \in \{1, 2, ..., 84\}$, $\Delta\phi_i, \Delta\theta_i, \Delta\psi_i \in [-5°, 5°]$. Among these 2,000 images, 1,600 are used for training and 400 are used for validation. An additional 200 images are generated in the same fashion for testing the success rate of our CNN.

We implemented the CNN using MatConvNet, leveraging models used initially for MNIST. Table III gives the parameters set for various layers of the CNN. Figure 7 shows the objective and error curves for this particular data set, the training process converges reasonably fast. When the trained CNN is applied to the test images, the rate of successful orientation prediction is 99.4%.

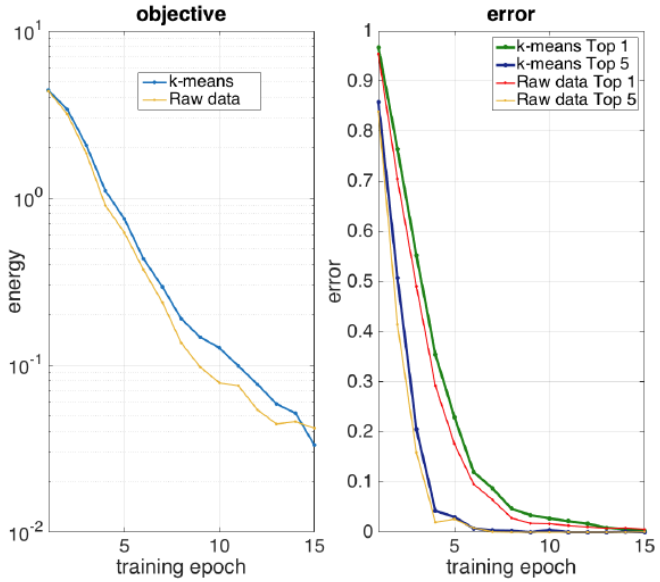


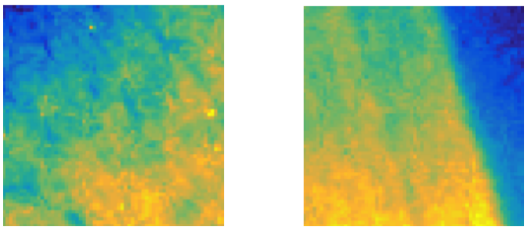
Fig. 7: Case I: Objective and Error curves from the training process, both using the original (raw) and the quantized (in eight levels) data, showing that the quantized data had very similar accuracy to the original one.

B. Case II: X-ray diffraction

To test the viability of using CNN to identify X-ray diffraction images with visible Bragg spots, we use an experimental data set collected at the Stanford Linac Coherent Light Source (LCLS) that contains diffraction patterns of a number of photosystem II crystals. The data set contains 7,066 images total. These images have been previous labeled by beamline scientists. Among them, 3,555 images present "good" Bragg peaks, and 3,511 images do not contain clearly defined Bragg peaks. Each diffraction pattern is represented by $1,920 \times 1,920$ pixel images.

To prepare the training data, we randomly selected 100 images from the set that contain Bragg peaks and 100 images from the set that contains images with poor qualities. We divide each image into 900 64×64 frames, and examined and labeled each one of these images by hand. Because Bragg spots are sparsely distributed in high quality diffraction images. Many frames extracted from a high quality diffraction image may not contain any Bragg spot.

The architecture of the CNN we use for training is shown in Table IV. Due to relatively small size of the data, only two convolution layers are used in this CNN. The training process converges quickly. When applied to the rest of the data, the CNN correctly identifies the presence and absence of Bragg peaks in 96.3% of the 64×64 frames. If a diffraction image contains at least 3 frames that contain Bragg spots, we consider that image to contain a high quality diffraction pattern with Bragg spots. By this metric, the CNN corrected identifies 100% images that contain Bragg spots.



(a) False negative (b) False positive

Fig. 8: Case II: Two 64×64 frames from the same diffraction image. The one on the left contains Bragg spots and the one on the right does not contain any visible Bragg spots.

TABLE IV: The architecture of the CNN used to identify X-ray diffraction images that contain good Bragg spots

Layer No.	Layer Type	Output Size	Filter Size
1	Input	64 x 64 x 1	—
2	Convolutional	18 x 18 x 100	11 x 11
3	Max-pooling	6 x 6 x 100	3 x 3
4	RELU	6 x 6 x 100	1 x 1
5	Convolutional	1 x 1 x 84	6 x 6
6	Fully connected	84 x 1	—

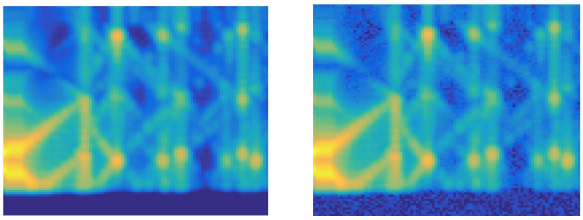
C. Case III: X-ray scattering

We address the problem of classifying GISAXS image patterns of 7 different crystal lattices, whose classes we label as: BCC 100, BCC 110, Cubic 100, Cubic 110, FCC 100, FCC 111, and HCP. We generated 1,000 images for each class (dataset 1), with dimensions of 100 $\times$ 100 pixels and variable lattice parameters, including particle sizes (10nm - 20nm) and spacings. The classifier is a four-layer CNN shown in Table VI. We also tested the performance of the classifier on dataset 2, which is generated with different simulation parameters, and an extra synthetic dataset by corrupting images in dataset 1 with Poisson noise at a realistic level (see Figure 9). We obtained test accuracy of 92% on dataset 1 and 82.6% on dataset 2, which indicates that the approach achieves generalization under noisy conditions. Also, the trained model on noisy data (dataset 2) will drop only 0.72% accuracy when used on dataset 1. We also implemented a linear classifier with features extracted by Histogram of Oriented Gradients (HOG) [31], which has been widely applied in various computer vision tasks. The CNN model outperforms HOG on all data sets, especially on corrupted images. The obtained results are shown in Table V. This is an encouraging result for further extending the use of CNNs for GISAXS as well as other synchrotron based scientific experiments.

We modified AlexNet network structure, but reduced the number of fully-connected layers to accelerate training. Training was run on an Intel(R) Xeon(R) CPU E5-2630 2.40GHz along with an Nvidia K-80 with 24GB of RAM and finished within 10 minutes.

TABLE V: Case III: Crystals classification accuracy on different test datasets

Method	dataset 1	dataset 2	dataset 1(noisy)
CNN	92.29	82.60	91.57
HOG	92.00	79.88	79.57



(a) A BCC 100 image without noise (b) The same image with noise

Fig. 9: Case III: Example of corrupting GISAXS images with Poisson noise

D. Case IV: X-ray attenuation contrast

In order to evaluate the performance of a CNN when classifying fiber profiles coming microCT images, we trained an AlexNet model with 70% of the samples and tested on the remaining 30%, given a dataset with 300,000 samples.

The features of the last layer, a fully-connected one, is input for an image retrieval scheme based on a distance. The number of parameters of all layers is 571,648 and the number of parameters in the fully-connected layer corresponds to 81.78% of the parameters to be estimated. Table VII shows the TensorFlow architecture including the output and filter size of each layer of the CNN, which was the architectural configuration of pyCBIR [32] version 1. pyCBIR is python package

TABLE VI: CNN architecture applied to X-ray scattering data.

Layer No.	Layer Type	Output Size	Filter Size
1	Input	100 x 100 x 1	—
2	Convolutional	96 x 96 x 20	5 x 5
3	Max-pooling	48 x 48 x 20	2 x 2
4	Convolutional	44 x 44 x 20	5 x 5
5	Max-pooling	22 x 22 x 20	2 x 2
6	Convolutional	19 x 19 x 50	4 x 4
7	Fully connected	7 x 1	—

TABLE VII: TensorFlow CNN architecture applied to fiber dataset.

Layer No.	Layer Type	Output Size	Filter Size
1	Input	16 x 16 x 1	—
2	Convolutional	16 x 16 x 64	5 x 5
3	Max-pooling	8 x 8 x 64	2 x 2
4	Convolutional	8 x 8 x 64	5 x 5
5	Max-pooling	4 x 4 x 64	2 x 2
6	Fully connected	382 x 1	—
7	Fully connected	192 x 1	—

that offers a search engine for user-friendly exploration of general image composition. The system enables image-based query to digital imagery archives, returning the top-n pictures that most resembles the query, for n defined by the user. Given, we obtained 99.788623% accurately retrieved searches, considering stratified training and tests sets, with 70% and 30% of the samples respectively.

Our search engine builds upon computer vision advances made over the past decades, including both low-level feature design and Alexnet CNNs. We demonstrate the applicability to microCT by classifying fiber cross-sections, automatically linking unseen results to known labeled patterns; it is also capable of recognizing patterns from sketched cartoons of fibers. Moreover, our pyCBIR [32] tools incorporate six feature extraction methods, ten similarity criteria and numerous ways to query and retrieve images. Ongoing and future extensions of this image search system are expected soon, and future developments appear in Section **??**.

V. Energy efficient CNN and neuromorphic computing

Currently, CNNs are often used on high performance CPUs and GPUs to solve a variety of data analytic problems. Although the latest advances and availability of fast processors and memory has made it possible to use machine learning algorithms and large volumes of data to identify patterns and perform classifications, the energy consumption required in these tasks has become a major concern in applications that require implementing CNN on digital devices or equipment.

One intriguing observation is that a CNN often produces a rather simple answer (e.g, yes or no or a class number), and yet the type of operations performed in CNN currently require at least 32-bit arithmetic.

The IBM TrueNorth chip is built on a neuromorphic model in which the processing unit and the memory are collocated. The input data is represented as spiking trains and processed in an event driven fashion. These features allow TrueNorth to process thousands of images with several Watts power budget.

By lowering the precision of the operations performed in a CNN, we are able to reduce the energy usage, and expect to operate from neuromorphic chips. In this section, we discuss the effect of input data and CNN weights quantization on the success rate of a CNN tested on cryo-EM. Next, we present preliminary results on using the IBM's TrueNorth chip in classifying microCT cross-sections.

A. Quantized CNN

To test the effect of limiting the precision of CNN operations, we took the CNN constructed for classifying 2D projection views of cryo-EM images in section III and quantized the convolution weight parameters. For example, to quantize a set of weights W into three levels *l1*, *l2*, and *l3*, we define δ_1 and δ_2 inside the range of W so that the quantized weights $\hat{\text{W}}$ is defined by:

$$\hat{W}_i = \begin{cases} l1 & \text{if } W_i < \delta_1 \\ l2 & \text{if } \delta_1 \leq W_i < \delta_2 \\ l3 & \text{if } \delta_2 \leq W_i \end{cases}$$

The values of *l1*, *l2*, and *l3* can be determined by using a clustering algorithm to select a cluster centroid among all the weights present in the CNN. We use K-means with Lloyds optimization [33].

A similar quantization scheme can be applied to the input data also. Figure 11 shows how the success rate of the quantized CNN changes with respect to the number of quantized levels. We can see that with a 2-bit quantization, i.e., representing each of the input pixel and CNN weigh by a single bit, we can still achieve 70-75% success rate. If we increase the number of levels to 8 (3-bit quantization), the success rate of the prediction is over 95%.

Figure 12 and Table VIII give a more detailed look at the effect of input data and CNN weight quantization on the success rate of the CNN. It is clear from these figures that we can indeed lower the precision of the input data and CNN weights without affecting the predictability of the CNN.

However, we should point out that here we use a train-then-constrain approach. The training process in which the CNN weights are optimized is still performed on GPUs and CPUs with full 32-bit floating point operations. It is only the deployed CNN we quantize. The quantized CNN can be represented by a few bits and a code book (a.k.a. lookup table.)

TABLE VIII: CNN success rate (%) using different levels of quantization for data, weights, and biases

Quant. levels	S. rate (%)
1	2,2
2	78,6
3	92,4
4	90,6
5	95,6
6	94,8
7	96,8
8	98,8

TABLE IX: CNN success rate (%) using different levels of quantization for data, weights, and biases

Q. levels	2^0	2^1	2^2	2^3
2^0	2,2	1,4	1	1
2^1	2,2	78,6	94,8	96,8
2^2	2,2	65,4	90,6	98,8
2^3	2,2	71,6	88,8	98,8

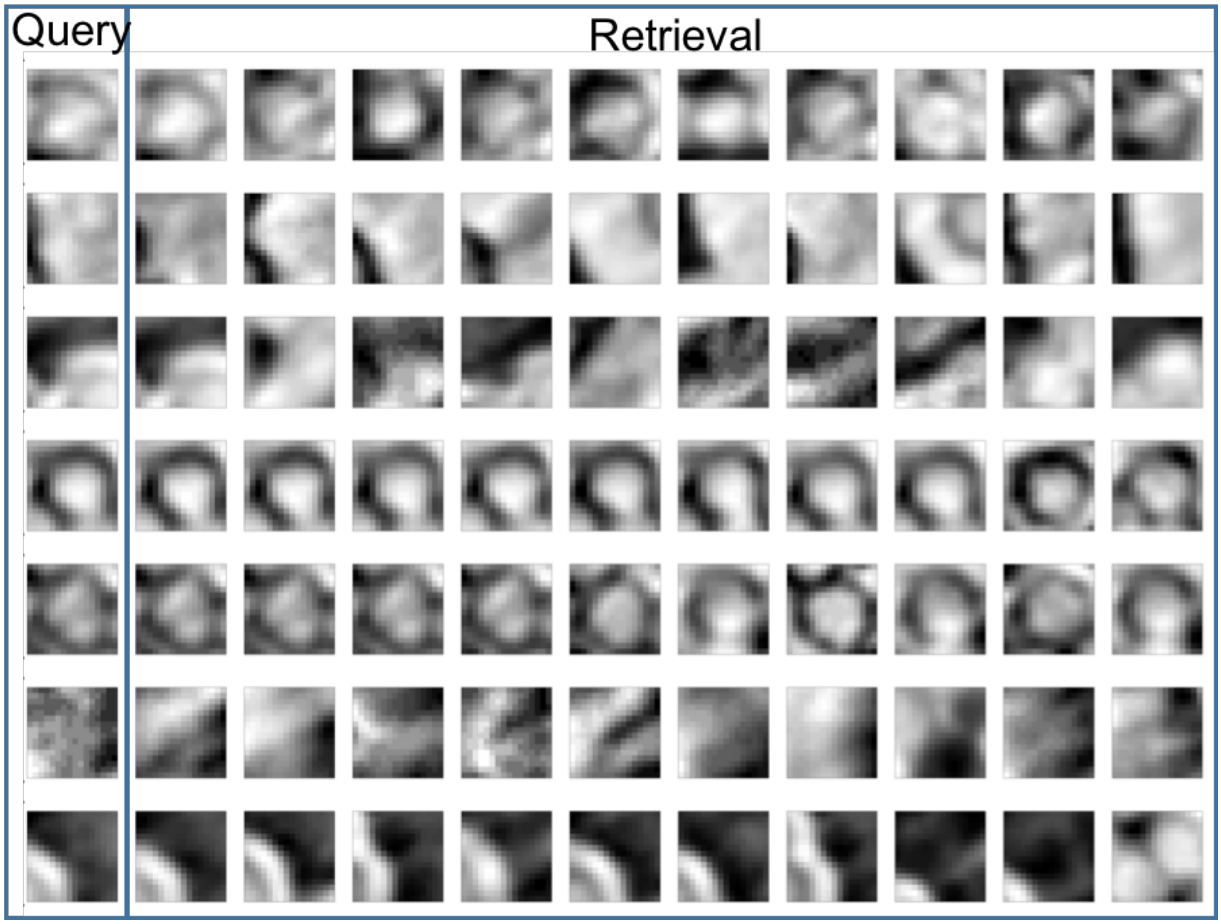


Fig. 10: Case IV: Using TensorFlow: Dataset with more than three hundred thousand samples, used to train a deep learning algorithm in order to separate fiber profiles (left) from other regions (right).

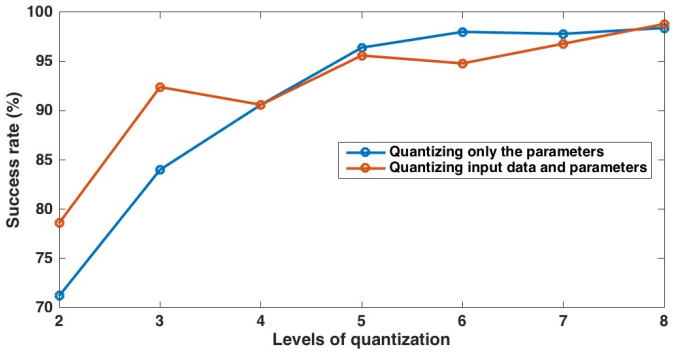


Fig. 11: Relation between the number of quantization levels and the success rate using cryo-EM images.

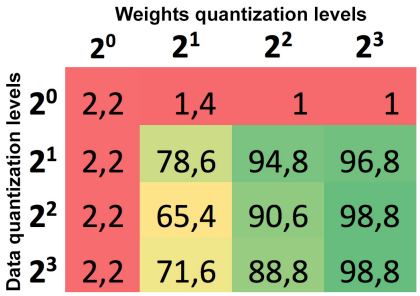


Fig. 12: CNN success rate (%) using different levels of quantization; colors are proportional to the accuracy rate.

B. *Experience with IBM TrueNorth*

Through Matlab, MatConvNet and Eedn, we deployed algorithms that could be trained on GPU clusters ($\approx 15 minutes$ on a TITAN GPU card, with 0.2 seconds per iteration), which produced a model that was ported to the IBM TrueNorth chip. By using the IBM TrueNorth ns1e at 240 ticks/frames, we achieved the following performance results in two different scenarios: (a) the run-time for 240 ticks was 0.250299 sec., and it takes 1.04 milliseconds to classify each frame/image; (b) by trying to overclock the chip to investigate if it can run even faster, we obtained 0.125363 sec run-time for 240 ticks, which takes up to 0.5 milliseconds per frame/image.

Preliminary results using the neuromorphic chip IBM TrueNorth utilized ≈3,200 cores, which corresponds to less than the amount of cores in one chip (4096 cores). The

classifications results showed accuracy of 99.788623% for 70% samples used in training and 30% for testing.

Other important specifications associated to our tests are: (a) the Corelet Programming Environment CPE Version 2.2.160518; (b) dataset with 216,650 fiber samples and 105,120 non-fiber samples, each sample consisting of 16x16 uint8 images.

VI. Conclusion and Discussions

This paper presented encouraging results towards the use of CNNs for classification of cryoEM and X-ray-based patterns. Since specialized imaging facilities like those at national laboratories collect vasts amounts of data, the use of CNNs for pattern recognition automation offers an opportunity to learn tasks from previously curated data sets. For example, CNN supported the analysis and inversion of new scattering patterns, otherwise a time consuming process that requires significant human expertise.

The use of CNNs as a core algorithm, as explored here, can automate much of the data analysis processing and accelerate the pace of scientific discovery. In addition, we realize that this algorithm works well in the reconnaissance of expected events, however it currently lacks the ability to detect outliers, which may be key to answer science questions about unique events buried in experimental observational data or simulation.

We notice that image representation can be arbitrary among different packages, i.e. the order in which the image dimensions appear varies. While testing different CNN architectures to different imaging modalities, we also noticed that the number of deep layers were proportional to the image resolution and complexity of the patterns. The ability to process higher resolution data demands larger convolutional layers as well as an increased number of them (deep), which constrain many CNN designs computationally because they require more memory as well as more computing time.

Exploration of new computer architectures with neuromorphic microelectronic promises to enable very efficient chips that can bring low-power computation closer to the data sources. We expect that predictive analytics will steer data collection, therefore providing more efficient data storage, i.e., data with high SNR remains, otherwise discarded early in the process. By having recognition closer to the detector, we expect to obtain higher resolution data in specific regions, or more time points during a transition of interest.

We will continue developing these software tools for the recognition of the scientific data collected at imaging facilities, including other modalities and materials [34], seeking to better categorize geometrical motifs present in images. The task of extracting morphological structures and representing them through features maps will have similarities to traditional pattern recognition, and we intend to build analogies and extend concepts to 3D imaging modalities. Advanced recognition will significantly accelerate the model development for abstract patterns, which are often hard to describe visually.

Acknowledgments

The authors would like to thank the grants LBNL LDRD Neuromorphic Computing for Image Recognition and the Early Career Research project IDEAL, both under ASCR (DE-AC02-05CH11231) and the Center for Applied Mathematics for Energy Related Applications (CAMERA), under management of ASCR and Office of Basic Energy Sciences (BES) of the U.S. Department of Energy. We also thank IBM Almaden, particularly Ben Shaw and Alexander Andreopoulos for supporting our team developments during the IBM TrueNorth Boot Camp in May of 2016. Any opinion, findings, and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of the Department of Energy.

References

[1] B. Tischleder and S. Wasserman, *Cultures of obsolescence*. Palgrave MacMillan, 2015.

[2] W. Bethel, M. Greenwald, and L. Nowell, "Management, visualization, and analysis of experimental and observational data (EOD) - the convergence of data and computing," in *DOE ASCR Workshop*, pp. 2–30, DOE, 2015.

[3] D. H. Ballard and C. M. Brown, *Computer Vision*. Prentice Hall Professional Technical Reference, 1st ed., 1982.

[4] R. C. Gonzalez and R. E. Woods, *Digital Image Processing*. Pearson, 3rd ed., 2007.

[5] A. Wills, D. Michalak, P. Ercius, E. Rosenberg, T. Perciano, D. M. Ushizima, R. Runser, and B. Helms, "Block copolymer packing limits and interfacial reconfigurability in the assembly of periodic mesoporous organosilicas," *Advanced Functional Materials*, pp. 1616–3028, Jun 2015.

[6] L. Zhang, H. P. H. Shum, and L. Shao, "Discriminative semantic subspace analysis for relevance feedback," *IEEE Transactions on Image Processing*, vol. 25, pp. 1275–1287, March 2016.

[7] J. Frank, *Three-Dimensional Electron Microscopy of Macromolecular Assemblies: Visualization of Biological Molecules in Their Native State*. Oxford University Press, 2006.

[8] A. Thompson, D. Attwood, E. Gullikson, M. Howells, K.-J. Kim, J. Kirz, J. Kortright, I. Lindau, Y. Liu, P. Pianetta, A. Robinson, J. Scofield, J. Underwood, G. Williams, and H. Winick, "LBNL advanced light source." http://xdb.lbl.gov/xdb.pdf, 2016. 2016-10-10.

[9] U. S. Department of Energy National Laboratory, "LBNL advanced light source." http://als.lbl.gov/, 2016. 2016-10-10.

[10] A. Hexemer, "Hipgisaxs: quick starter guide." https://github.com/HipGISAXS/HipGISAXS. 2016-10-10.

[11] H. A. Bale, A. Haboub, *et al.*, "Real-time quantitative imaging of failure events in materials under load at

temperatures above 1,600C.," *Nat Mater*, vol. 12, pp. 40–46, 2012.

[12] D. Ushizima, T. Perciano, H. Krishnan, B. Loring, H. Bale, D. Parkinson, and J. Sethian, "Structure recognition from high resolution images of ceramic composites," *IEEE International Conference on Big Data*, Oct. 2014.

[13] D. M. Ushizima, H. A. Bale, E. W. Bethel, P. Ercius, B. A. Helms, H. Krishnan, L. T. Grinberg, M. Haranczyk, A. A. Macdowell, K. Odziomek, D. Y. Parkinson, T. Perciano, R. O. Ritchie, and C. Yang, "Ideal: Images across domains, experiments, algorithms and learning," *The Journal of The Minerals, Metals & Materials Society*, pp. 1–10, Sept. 2016.

[14] J. Schindelin, I. Arganda-Carreras, E. Frise, V. Kaynig, M. Longair, T. Pietzsch, S. Preibisch, C. Rueden, S. Saalfeld, B. Schmid, J.-Y. Tinevez, D. J. White, V. Hartenstein, K. Eliceiri, P. Tomancak, and A. Cardona, "Fiji: an open-source platform for biological-image analysis," *Nat Meth*, vol. 9, pp. 676–682, July 2012.

[15] M. Hubel and T. N. Wiesel, *Brain and Visual Perception*. Oxford Univeristy Press, 2005.

[16] R. Ana, S. Esser, H. Simon, and D. Modha, "The cat is out of the bag: cortical simulations with 10^9 neurons, 10^{13} synapses," *Proceedings Super Computing*, pp. 1–12, 2009.

[17] T. M. Mitchell, *Machine Learning*. New York, NY, USA: McGraw-Hill, Inc., 1 ed., 1997.

[18] J. J. Hopfield, "Neural networks and physical systems with emergent collective computational abilities," *Proceedings of the National Academy of Sciences of the United States of America*, vol. 79, pp. 2554–2558, Apr. 1982.

[19] D. E. Rumelhart, G. E. Hinton, and R. J. Wilson, "Learning representations by back-propagating errors," *Nature*, vol. 323, pp. 533–536, 1986.

[20] A. Saxena, "Convolutional neural networks (cnns): An illustrated explanation." http://xrds.acm.org/blog/2016/06/convolutional-neural-networks-cnns-illustrated-explanation/, 2016. 2016-10-10.

[21] B. V. K. A. Mahalanobis and D. Casasent, "Minimum average correlation energy filters," *Applied Optics*, vol. 26, pp. 3633–3640, 1987.

[22] B. V. Kumar, A. Mahalanobis, and A. Takessian, "Optimal tradeoff circular harmonic function (otchf) correlation filter methods providing controlled in-plane rotation response," *IEEE Transactions on Image Processing*, vol. 9, pp. 1025–1034, 2000.

[23] VLFEAT, "Matconvnet." http://www.vlfeat.org/matconvnet/training/, 2016. 2016-10-10.

[24] A. Vedaldi and K. Lenc, "Matconvnet – convolutional neural networks for matlab," in *Proceeding of the ACM Int. Conf. on Multimedia*, 2015.

[25] P. Merolla, R. Appuswamy, J. V. Arthur, S. K. Esser, and D. S. Modha, "Deep neural networks are robust to weight binarization and other non-linear distortions," *CoRR*, vol. abs/1606.01981, 2016.

[26] S. K. Essera, P. A. Merollaa, J. V. Arthura, A. S. Cassidya, R. Appuswamya, A. Andreopoulosa, D. J. Berga, J. L. McKinstrya, T. Melanoa, D. R. Barcha, C. di Nolfoa, P. Dattaa, A. Amira, B. Tabaa, M. D. Flicknera, and D. S. Modhaa, "Convolutional networks for fast, energy-efficient neuromorphic computing," *Proceedings of the National Academy of Sciences*, vol. 113, pp. 11441–11446, Oct. 2016.

[27] D. S. Modha, "Introducing a brain-inspired computer: Truenorth neurons to revolutionize system architecture." http://www.research.ibm.com/articles/brain-chip.shtml, 2014. 2016-10-10.

[28] M. Abadi, A. Agarwal, P. Barham, E. Brevdo, Z. Chen, C. Citro, G. S. Corrado, A. Davis, J. Dean, M. Devin, S. Ghemawat, I. J. Goodfellow, A. Harp, G. Irving, M. Isard, Y. Jia, R. Jzefowicz, L. Kaiser, M. Kudlur, J. Levenberg, D. Mane, R. Monga, S. Moore, D. G. Murray, C. Olah, M. Schuster, J. Shlens, B. Steiner, I. Sutskever, K. Talwar, P. A. Tucker, V. Vanhoucke, V. Vasudevan, F. B. Vigas, O. Vinyals, P. Warden, M. Wattenberg, M. Wicke, Y. Yu, and X. Zheng, "Tensorflow: Large-scale machine learning on heterogeneous distributed systems.," *CoRR*, vol. abs/1603.04467, 2016.

[29] Google, "TensorFlow." https://www.tensorflow.org/, 2016. 2016-10-10.

[30] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," in *Advances in Neural Information Processing Systems 25* (F. Pereira, C. J. C. Burges, L. Bottou, and K. Q. Weinberger, eds.), pp. 1097–1105, Curran Associates, Inc., 2012.

[31] N. Dalal and B. Triggs, "Histograms of oriented gradients for human detection," in *Proceedings of the 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05) - Volume 1 - Volume 01*, CVPR '05, (Washington, DC, USA), pp. 886–893, IEEE Computer Society, 2005.

[32] D. Ushizima, F. Araujo, and R. Silva, "Searchable datasets in python: images across domains, experiments, algorithms and learning." http://pydata.org/sfo2016/schedule/presentation/14/, 2016. 2016-10-10.

[33] S. Lloyd, "Least squares quantization in pcm," *IEEE Transactions on Information Theory*, vol. 28, pp. 129–137, Mar 1982.

[34] A. W. Wills, D. J. Michalak, P. Ercius, E. R. Rosenberg, T. Perciano, D. Ushizima, R. Runser, and B. A. Helms, "Block copolymer packing limits and interfacial reconfigurability in the assembly of periodic mesoporous organosilicas," *Advanced Functional Materials*, pp. 1–9, 2015.