JOURNAL OF SYNCHROTRON RADIATION
ISSN 1600-5775

A convolutional neural network approach to calibrating the rotation axis for X-ray computed tomography

Xiaogang Yang,[a]* Francesco De Carlo,[a] Charudatta Phatak[b] and Doğa Gürsoy[a]

[a]X-ray Science Division, Argonne National Laboratory, 9700 South Cass Avenue, Lemont, IL 60439, USA, and [b]Materials Science Division, Argonne National Laboratory, 9700 South Cass Avenue, Lemont, IL 60439, USA. *Correspondence e-mail: yangx@anl.gov

Received 19 October 2016
Accepted 18 December 2016

Edited by J. F. van der Veen

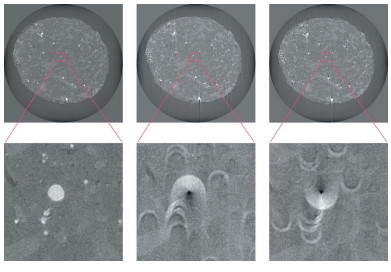
Keywords: tomography reconstruction; rotation axis; convolutional neural network; open-source; Python.

This paper presents an algorithm to calibrate the center-of-rotation for X-ray tomography by using a machine learning approach, the Convolutional Neural Network (CNN). The algorithm shows excellent accuracy from the evaluation of synthetic data with various noise ratios. It is further validated with experimental data of four different shale samples measured at the Advanced Photon Source and at the Swiss Light Source. The results are as good as those determined by visual inspection and show better robustness than conventional methods. CNN has also great potential for *reducing or removing* other artifacts caused by instrument instability, detector non-linearity, *etc*. An open-source toolbox, which integrates the CNN methods described in this paper, is freely available through GitHub at tomography/xlearn and can be easily integrated into existing computational pipelines available at various synchrotron facilities. Source code, documentation and information on how to contribute are also provided.

1. Introduction

X-ray computed tomography (XCT) scans at today's synchrotron light sources can yield thousands of image frames per second at high resolution (Finegan *et al.*, 2015; Cai *et al.*, 2016; Mader *et al.*, 2011; Xiao *et al.*, 2012; Moosmann *et al.*, 2013). Typical data volumes from a single scan are of the order of tens of gigabytes; however, for larger specimens this number can be up to three orders of magnitude larger (Atwood *et al.*, 2015). Moreover, data generation rates will significantly increase after the upgrade of the storage rings that are planned or under development at many synchrotron facilities worldwide (Relch, 2013; Castelvecchi, 2015). Current and expected data volumes and rates necessitate having reliable, efficient and fully automated data processing pipelines. One major bottleneck in XCT automation is the poor calibration of the center-of-rotation (CoR) with the available numerical algorithms.

Current methods for CoR calibration are mostly heuristic. One common method is by shifting each image such that the center-of-mass in each projection image is well aligned (Azevedo *et al.*, 1990). Another is to use two projection images taken at rotations that are 180° apart (Yang *et al.*, 2015). Both have been demonstrated to be effective for good quality datasets, but they tend to fail for increased noise levels. There are other methods that are based on advanced calculations, such as Fourier analysis (Vo *et al.*, 2014) and feature detection routines (Yang *et al.*, 2015), or based on solving an optimization problem (Donath *et al.*, 2006). None of these methods have proven to provide reliable calibration for a wide range of



data and noise levels. The manual estimation of CoR by visual inspection is still the most reliable way, but makes the automation of data processing not possible.

Machine learning provides a robust solution to emulate human intelligence by learning from existing model data relationships, and is particularly suitable for analysis of high-dimensional complex data problems (Pereira *et al.*, 2009). Convolutional Neural Network (CNN) (LeCun *et al.*, 1998) is a popular machine learning technique for image processing. It is capable of classifying structural features in images similar to the way the human visual system operates. Its accuracy and efficiency for feature recognition and classification have been proved for various applications (Mirowski *et al.*, 2009; Lawrence *et al.*, 1997; Garcia & Delakis, 2004; Matsugu *et al.*, 2003).

In this paper, we present a CNN classification model to calibrate the CoR for X-ray tomography. Starting from the basics of CNN, we develop an algorithm to automatically distinguish the reconstructions of correct and incorrect CoRs. We build an open-source software to implement this algorithm and present the implementation details. We evaluate the algorithm with both synthetic data for different noise levels and experimental data from different synchrotron facilities. The results are compared with conventional methods. We also discuss the future development of the CNN for synchrotron imaging problems.

2. Methods

Well centered and off-centered reconstructions can be directly determined by visual inspection manually without any numerical algorithm. This provides the most accurate way to evaluate the results for a final step. However, this approach is not practical for large datasets. Here we propose, instead, the use of a CNN classification model to emulate the process of the human brain. Thus, an automatic routine to compute the tomographic rotation axis is developed.

The rotation axis problem can be considered as an image classification problem, because there are significant different features between well center and off-centered reconstructions. We *train* the CNN to distinguish the different features with a few manually classified images and use the trained CNN to automatically find the correct rotation center.

2.1. Background

Among artificial neural networks, CNN provides a model to learn representations of data with multiple levels of abstraction akin to human visual processing (LeCun *et al.*, 1998). Broadly speaking, it is a unique process to build a function f between the input data X and output data Y ($f : X \rightarrow Y$), which is the so-called supervised learning. This f is not like the traditional formulas of representing the data relations with simple mathematical operators and symbols. It is a composition of multiple layers of weights (W) with activation functions (K). After we fit W for specific input and output data ($X \rightarrow Y$), this f can be used to predict the future data with the same rule of the fitting data. The specially designed CNN architecture can work as unsupervised learning; however, in this article, we focus on the supervised learning architecture.

CNN was originally developed for image classifications. Its basic and most popular applications are hand-writing recognition and human face recognition. In these cases, CNN plays the role of a fitting function between the input images and the output labels. The process to fit W of the CNN model is called *train*. The iterations during *train* are called *epochs*. Typically, stochastic gradient descent (SGD) based methods are used for training. We use a popular one of these, called *Adam* (Kingma & Ba, 2014). Once the CNN is trained for a specific data model, we can use it as the function to estimate the label of an unknown image containing features that are close to the training data. This step is called *predict*.

Besides the basic structures of the conventional artificial neural networks (Basheer & Hajmeer, 2000), CNN includes two unique hidden layers: a convolution layer and a sub-sampling layer. The convolution layer uses a small convolution matrix to compute the convolution of the image, which works as the image filter. One convolution layer contains tens to hundreds of convolution matrices for extracting multiple features of the same image. The convolution layer works as a feature bank. The convolution matrix (kernel), which is randomly generated at the initial iteration, is the W to be fitted for CNN. The sub-sampling layer follows the convolution operation to reduce the image size with specific ratio through a form of non-linear down-sampling. Max-pooling is a popular method of sub-sampling, in which the input image is partitioned into a set of non-overlapping rectangular regions, and for each such sub-region we pick the maximum value. The sub-sampling layer makes the image information sparse enough to be fitted with a simple value of the output label.

2.2. CNN configuration

There is not a standard CNN architecture for image classification. After we tested different architectures and parameters to consider their performances and stability, we chose to use the CNN architecture as shown in Fig. 1. This includes three convolution layers and two max-pooling layers. The first convolution layer includes 32 convolution weights to extract 32 feature maps from the inputs. The image size reduces to half at each max-pooling layer. The number of convolution weights and feature maps doubles after each max-pooling layer. The final layer of the feature maps are fully connected to data nodes with the activation function. These nodes are connected again with another activation function and become a single value. We fit this value to be the label that was defined in the training data.

The complicated architecture of CNN requires a high computing capability to process the training and prediction. A single central processing unit (CPU) of the desktop or workstation is too slow for this case. However, the CNN is very suitable for parallel computing, because there are many repeating computings of convolution or max-pooling on each layer. A graphical processing unit (GPU) workstation is recommended to implement the CNN efficiently.

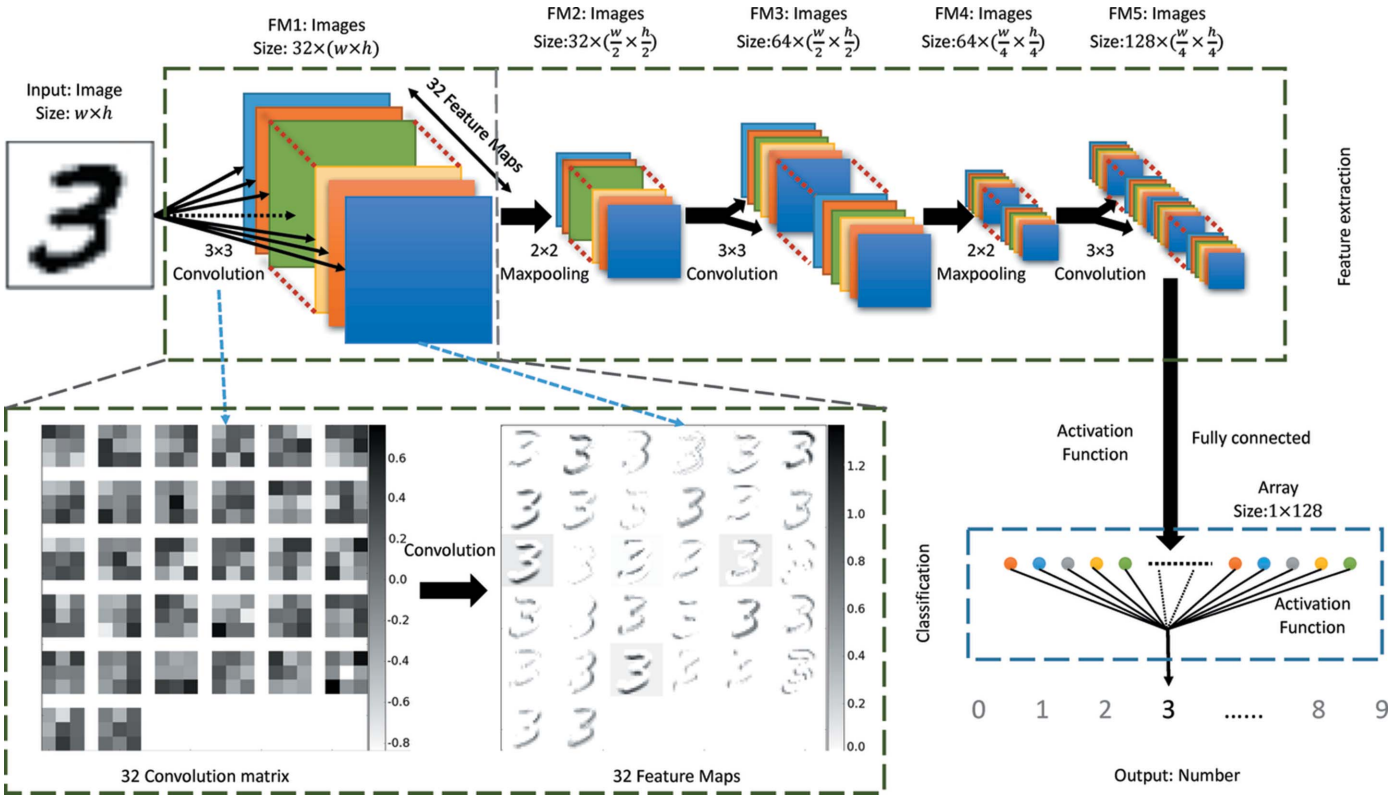


Figure 1
The architecture of the CNN classification model for this article. We use the classification of handwriting number 3 as the example. This diagram shows how the handwriting image has been classified as its related number.

The process to prepare the training data decides the computing model. The more training data we prepare, the better the prediction results will be. Normally the number of training data should be at least of the magnitude of 10^4 for a reasonable prediction (Krizhevsky *et al.*, 2012). This procedure is always considered difficult, because most of the steps for this task have to be performed manually. In some cases, like solving a general image classification problem for nature images, this can be an overwhelming task and explains why machine learning techniques are not yet widely applied. However, for the image classification problems of synchrotron imaging, the image features are normally restricted to some specific aspects and therefore we only need to use a few images to train the CNN model.

To dismiss the difference of the pixel value range from different tests or measurements, we regularize the pixel values before preparing the training data or testing data. The regularization equation is $I_r = (I - \bar{I})/\sigma_I$, where I is the pixel value of the input image, $\bar{I}$ is the mean pixel value, σ_I is the standard deviation of the pixel value and I_r is the regularized pixel value.

In the procedure of preparing the training data, we reconstruct a tomographic slice with various values of CoR. During the training phase, we select the well centered reconstruction by eye, and label the rest as off-centered reconstructions. We extract overlapped patches from the well centered and the off-centered images and label them 1 and 0, respectively.

A patch is a square window ($s_p \times s_p$) extracted from the image. The patches are overlapped one by one. The distance of the centers between two neighbor patches is the patch step (n_s). The patch number is $N_p = (1/n_s^2)(h - s_p)(w - s_p)$ for an image with height (h) and width (w). There are two reasons for using small patches instead of the whole image:

(i) We can generate sufficient train data from only one single image.

(ii) The overlapped patches provide multiple evaluations of the same feature of the image.

Once we have extracted the patches from the well centered and off-centered images, we select a specific number (P_{train}) of patches with their labels (Y_1) from both of these groups. We use the patches as input X_{train} and the labels as output to train the CNN model. The trained CNN classification model is now capable of distinguishing the well centered or off-centered patches.

The prediction procedure evaluates Y_1 of the patches from the reconstructions of different CoR. We choose a slice with typical features of the sample for a rapid evaluation. We first carry out tomographic reconstructions with various values of CoR in a specific range (*e.g.* ± 100 pixels around the desired CoR) for this slice. For each reconstructed image, we extract a specific number of patches. The size of the patches should be the same as the training data. The number of patches can be roughly hundreds for each reconstruction. We use these patches as the input data for trained CNN and predict their

label. If the value of the label is close to 1, it means the feature of the patch is close to being well centered. If it is 0, it is off-centered. We compute the sum (S_l) of the labels for the patches from one reconstruction. The reconstruction with the maximum S_l is the well centered one as our evaluation model.

2.3. Implementation details

We developed a Python toolbox named *Xlearn* (https://github.com/tomography/xlearn) to implement all the functions described above. The toolbox is currently based on the *Keras* (https://github.com/fchollet/keras.git) and the *Theano* (https://github.com/Theano/TheaNo.git) packages. *Keras* is a popular platform for artificial neural networks. *Theano* is a popular platform for tensor flow computing. These platforms all include GPU acceleration, which is the key feature to apply CNN on large datasets. As for our previous tests, the GPU can be hundreds of times faster than CPU to train the CNN model, allowing CNN to be tested and implemented for real cases.

The *Xlearn* toolbox includes two core functions of the CNN: image classification and image transformation. The architecture of the classification model is the same as that shown in Fig. 1. The package also includes the necessary accessories of normalizing the image data, extracting the patches from images and reconstructing images from the patches.

There are demonstration codes and related data to test the calibration of CoR for the shale samples which we will discuss in §3. There are also demonstrations using CNN to reduce ring artifacts and to segment the fluorescence image of biological cells. Detailed API references and examples can be found on the documentation website (http://xlearn.readthedocs.io/).

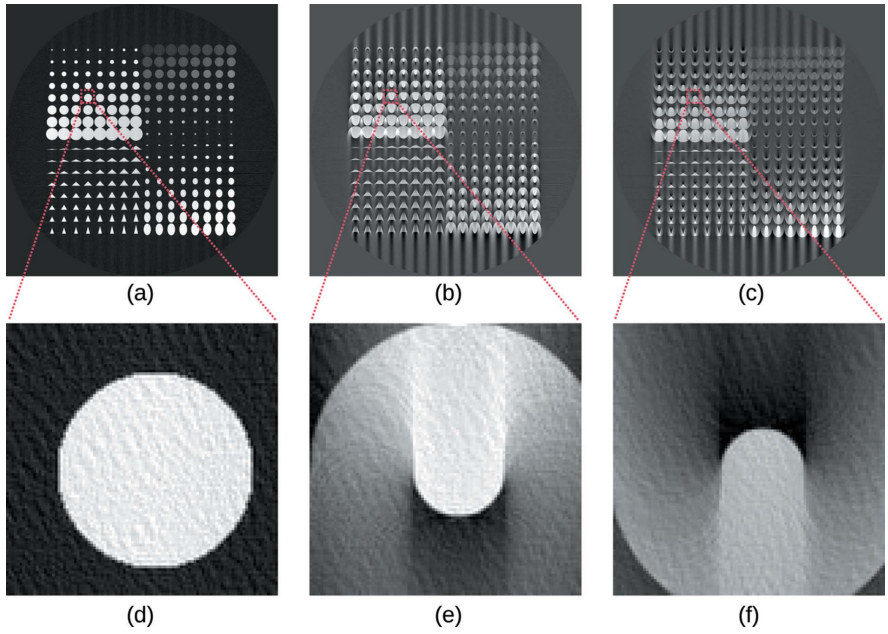


Figure 2
Artificially created training datasets include circles, ellipses and triangles of different size and gray value. The images reconstructed with correct (*a*) and incorrect (*b*, *c*) rotation axis are shown in the top row. Bottom row images (*d*, *e*, *f*) are zoomed-in sections of the reconstructed images.

3. Results

3.1. Evaluation with synthetic data

We generated a synthetic test image to train the CNN classification model. The size of the image is selected as 2900×2900 pixels, which is typical for full-field imaging experiments at synchrotrons. The training images were formed using basic geometric entities such as ellipses and triangles, because many structural features can be composed using a superposition of this basic set of elements. We simulated data according to Beer's law using 900 tilt angles from 0 to 180° using different noise levels, and performed a multitude of tomographic reconstructions using the original as well as different shifts for the rotation axis. This provided one correct and 200 off-centered reconstructed images. The shift of rotation center between different reconstructions is selected as half a pixel.

Fig. 2 shows the resulting true and off-centered reconstructed images used for training. All images were obtained using *GridRec* of the TomoPy toolbox (Gursoy *et al.*, 2014), which is the common reconstruction algorithm in synchrotron tomography. We extracted 100×100 patches randomly as the testing data from each reconstructed dataset and labeled them either correct or incorrect. We used these patches as input X_{train} and their labels as output Y_{train} to train the CNN classification model.

We then created a set of randomly generated images for evaluation of the trained network. Fig. 3 shows reconstructed images of a single sample for different noise levels. The box plot shows the accuracy of all samples for different noise levels. The method provided perfect rotation axes for all hundred noise-free datasets. The prediction mean is observed to be less than a pixel for noise levels as large as 30%; however, the standard deviation of the estimation drops exponentially as the noise ratio is increased linearly. Therefore, to estimate accurate results for noisier datasets, one should use multiple slices for obtaining a reliable rotation axis estimation.

3.2. Validation with experimental data

We used three experimental datasets from TOMCAT at the Swiss Light Source (SLS) and two datasets from beamline 2BM of the Advanced Photon Source (APS) (Kanitpanyacharoen *et al.*, 2013) for experimental validation. The noise and resolution characteristics of these datasets are slightly different from each other. We trained the network using the reconstructed images shown in Fig. 4, which is from SLS. The trained CNN is then used to predict the rotation axis for other datasets. Reconstructions with correct rotation axes are

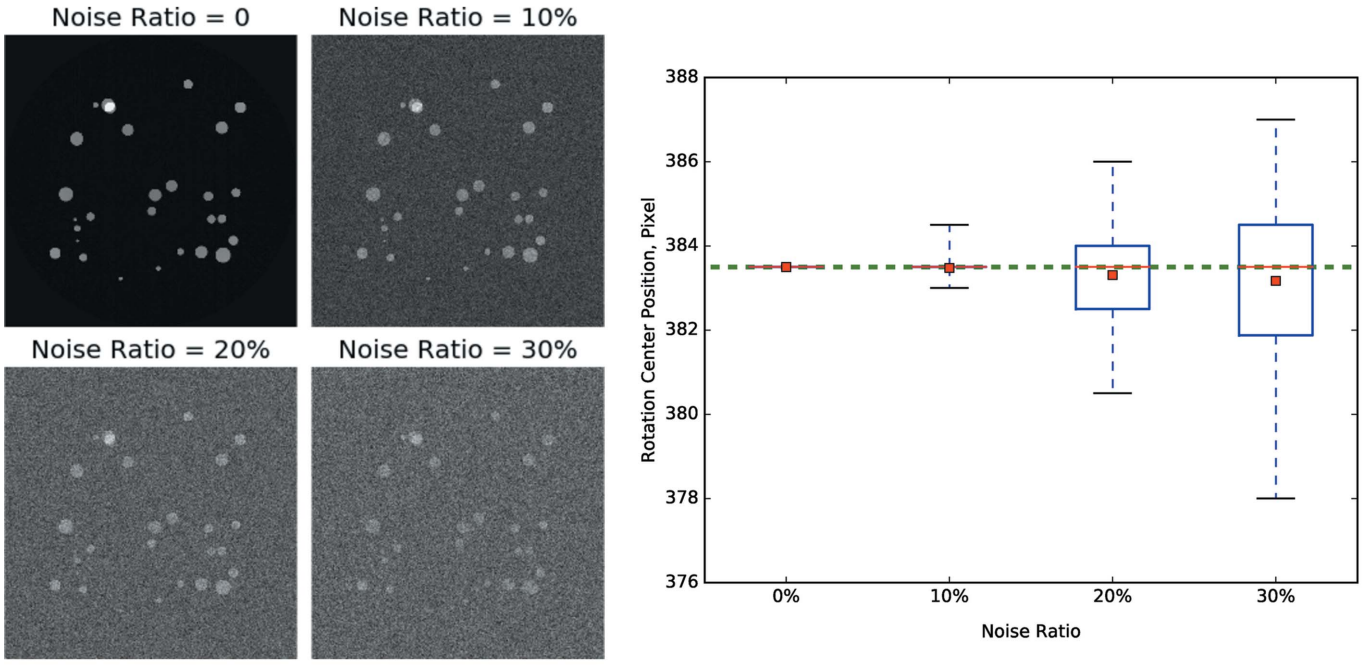


Figure 3
Statistical evaluation of the rotation centers predicted by CNN. The box plots shows the prediction statistics of the method for four different noise levels. The dashed green line shows the ground truth.

provided in Fig. 5. These results are exactly as we determined manually by eye.

To further evaluate the reliability of the algorithm, we use the trained CNN to predict the CoR of case 3 [Fig. 5(*c*)] for 50 different slices. The evaluate step for the rotation axis is 0.1 pixel. We also evaluate this case with the image registration based method (Manuel *et al.*, 2008) and the entropy based method (Donath *et al.*, 2006).

The statistics of the results are shown in Fig. 6. The results of the registration based method show a 0.5 to 1 pixel shift from the actual results that we estimated by eye. A few results show a large error of up to 6 pixels. The overall results are acceptable for estimating the rotation center with reasonable accuracy. The entropy based method fails for more than a quarter of the slices. The errors of the failed cases are up to 91 pixels, demonstrating that we should be cautious of using this method to blindly predict the rotation center axis. The mean value predicted by CNN is exactly the same as our estimation by eye. The median value has a 0.1 pixel shift. The majority of the results are in the range ± 0.2 around the actual results we estimated by eye. The maximum error is only 0.8 pixel. Overall, CNN shows reliable accuracy for the multiple slices of this case.

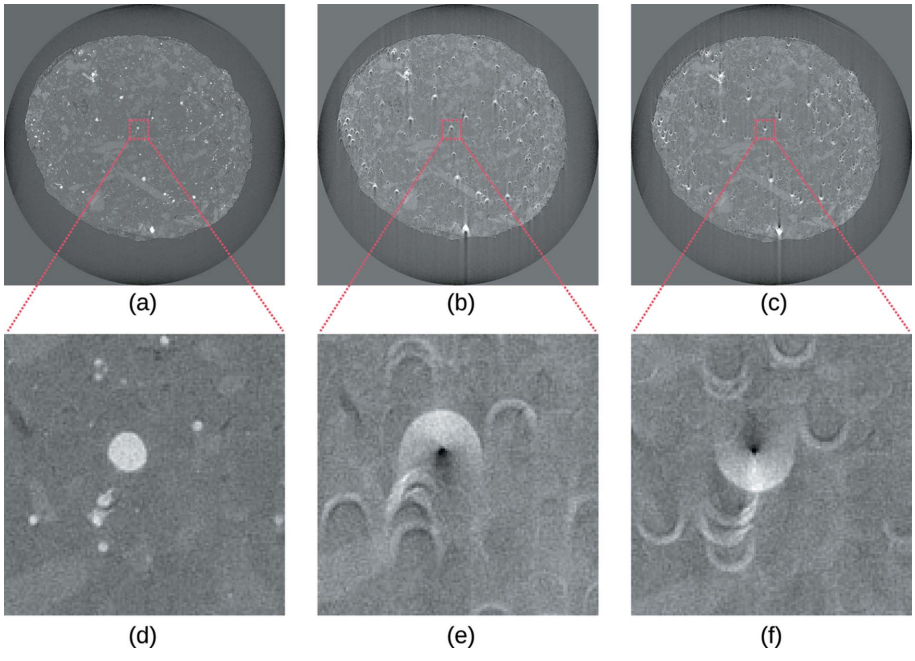


Figure 4
Samples of training data for experimental evaluation. (*a*) Well centered reconstruction. (*b*, *c*) Off-centered reconstructions. (*d*, *e*, *f*) Examples of patches extracted from different reconstructions. The reconstruction image size is 2048 × 2048 pixels. The patch image dimension is 100 × 100 pixels. These data are measured by TOMCAT at the Swiss Light Source.

4. Discussions

As shown in the tests above, CNN can predict the CoR with an accuracy as good as human estimation once we only provide one manually estimated training dataset. This can be a useful tool when we have a large number of

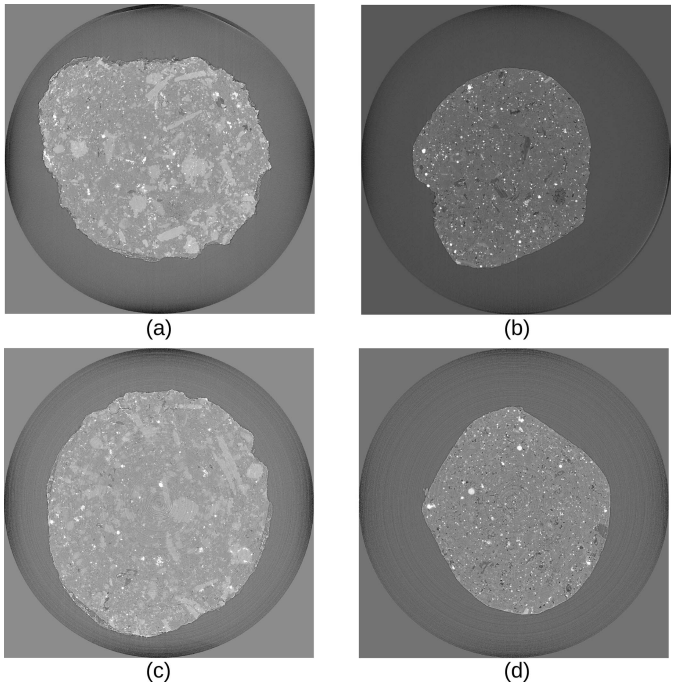


Figure 5
Four different well centered images that calibrated the CoR with CNN. (*a*, *b*) Slices of rock samples measured by TOMCAT at the Swiss Light Source. (*c*, *d*) Slices of the same rock sample measured at Advanced Photon Source beamline 2BM.

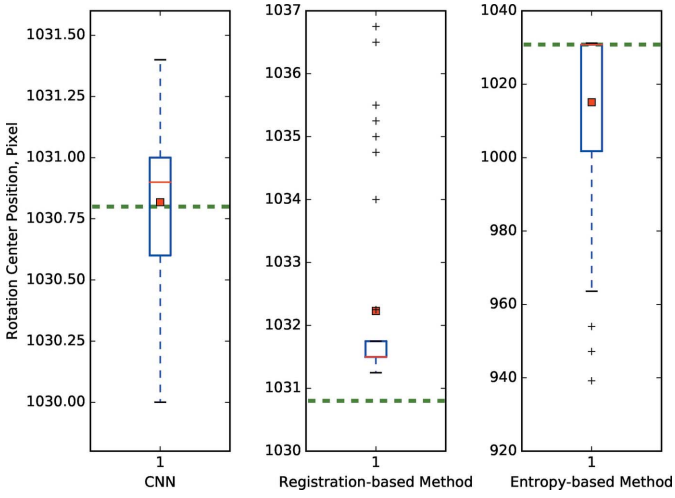


Figure 6
Rotation axes predicted by different methods using 50 slices of shale data as input. The CNN, mirrored image registration based method and entropy based method are shown from left to right. The dashed green line represents the ground truth.

tomographic measurements of similar samples. We only need to calibrate the CoR for one of the datasets and train the CNN to calibrate the rest of the datasets. The trained CNN model can be saved to a simple file. It can be easily loaded for the prediction process of future use. The prediction process using our *Xlearn* toolbox consists of a single line of Python code so it is simple to add to any automatic tomographic reconstructions and, as shown earlier, is as reliable as the manually selected CoR.

The computational efficiency of CNN makes it possible to process very large datasets. We tested our software on a workstation with an Intel Xeon E5-1660 processor, 64 GB memory and one NVIDIA Quadro M5000 GPU. For the cases above, we trained the CNN in 2 min. Once the network is trained, the evaluation of the CoR takes about 10 s per image. Computations for each image are independent, therefore the implementation can be easily scaled to meet a desired data throughput depending on the scan system used.

One thing we need to be cautious of using CNN for CoR calibration is that the data for the CoR calibration should have similar features as the training datasets. For instance, in the synthetic evaluation in this article we trained the CNN with the image including circles, ellipses and triangles. We obtain good results if we calibrate the CoR of tomographic images with these patterns but this might fail for totally different patterns. However, as shown in all our tests, the shape/size of the pattern and the quality of the image are not required to be the same in the training data and the data need to be calibrated. Because it is difficult to quantify the similarity between the training data and testing data, a guideline of preparing the training data cannot be built in the current stage. We suggest to prepare training data that contains the same types of pattern or feature as the testing data.

The CNN is robust in learning the way of human estimation for image analysis. To extend it for broader applications, we are developing a CNN architecture that directly builds a $f : X \rightarrow Y$ between two images. This step is called *image transformation* and we have already included it in the *Xlearn* toolbox. Image transformation has great potential for various imaging problems such as the ring artifacts correction, images segmentation, *etc*. Fig. 7 shows a simple demonstration of the basic idea, and recent results, not presented in this paper, show great promises in eliminating artifacts from reconstructions. Similarly, feature segmentation is often required for quantification of the structure, and CNN models can carry out this task quite efficiently. Based on these early results, we plan to expand the use of CNN to other synchrotron imaging problems present not only in tomographic data, and further carry out more detailed evaluations for different use cases.

In our test we used data available at the APS but, in general, assessing the robustness of new algorithms, not only CNN, requires access to a large library of tomographic datasets. Various efforts, starting from the ability to exchange data across facilities (De Carlo *et al.*, 2014), are underway aiming to create such a library. We wish that *Xlearn*, as a readily available tool box, will be continuously validated and expanded over time by users and experimentalists.

5. Conclusions

CNN is a robust approach to solving rotation axis determination in tomographic imaging. We developed a CNN classification model to calibrate the CoR that has been demonstrated to work accurately, once trained with sufficient prior estimation. Evaluation with synthetic phantoms shows good accuracy for different noise conditions. The validation

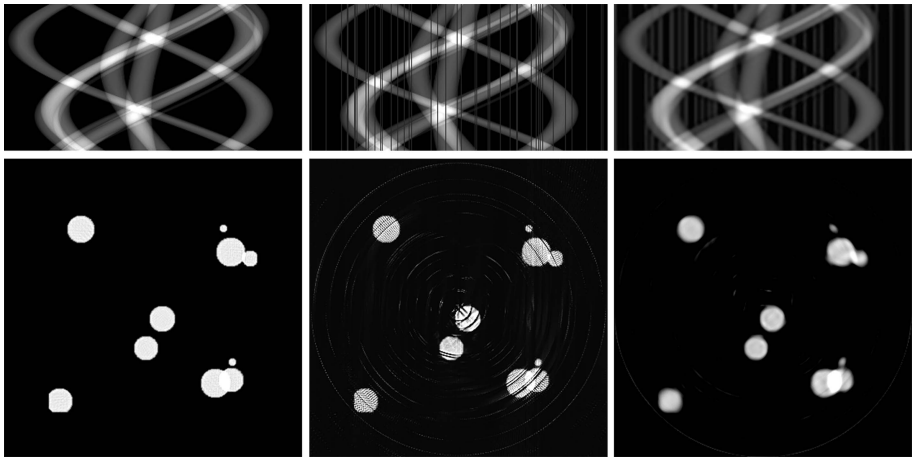


Figure 7
Demonstration of a potential other application of the CNN approach for ring artifacts correction. This case requires a different CNN architecture to transform the image from one style to another, which follows the same transformation rule as the training data. Image transformation of the *Xlearn* toolbox introduced in this article can be directly applied for this use. The top row shows the sinograms: (left) original, (middle) with added stripes, (right) stripes removed with CNN. The bottom row shows the corresponding reconstructions.

with experimental data is demonstrated by using a CNN trained with SLS data to reliably find the correct CoR for different shale samples measured at both the APS and the SLS. It also shows better accuracy than the image registration based method and the entropy based method for different slices of the shale sample. The CNN approach can calibrate the CoR for large datasets automatically as reliable as by human eye. Specially designed CNN architecture for image transformation is also promising to reduce ring artifacts and to segment images.

Acknowledgements

This research used resources of the Advanced Photon Source, a US Department of Energy (DOE) Office of Science User Facility operated for the DOE Office of Science by Argonne National Laboratory under Contract No. DE-AC02-06CH11357.

References

Atwood, R. C., Bodey, A. J., Price, S. W. T., Basham, M. & Drakopoulos, M. (2015). *Philos. Trans. R. Soc. A*, **373**, 20140398.
Azevedo, S. G., Schneberk, D. J., Fitch, J. P. & Martz, H. E. (1990). *IEEE Trans. Nucl. Sci.* **37**, 1525–1540.
Basheer, I. A. & Hajmeer, M. (2000). *J. Microbiol. Methods*, **43**, 3–31.
Cai, B., Lee, P. D., Karagadde, S., Marrow, T. J. & Connolley, T. (2016). *Acta Mater.* **105**, 338–346.
Castelvecchi, D. (2015). *Nature (London)*, **525**, 15–16.
De Carlo, F., Gürsoy, D., Marone, F., Rivers, M., Parkinson, D. Y., Khan, F., Schwarz, N., Vine, D. J., Vogt, S., Gleber, S.-C., Narayanan, S., Newville, M., Lanzirotti, T., Sun, Y., Hong, Y. P. & Jacobsen, C. (2014). *J. Synchrotron Rad.* **21**, 1224–1230.
Donath, T., Beckmann, F. & Schreyer, A. (2006). *J. Opt. Soc. Am. A*, **23**, 1048–1057.
Finegan, D. P., Scheel, M., Robinson, J. B., Tjaden, B., Hunt, I., Mason, T. J., Millichamp, J., Di Michiel, M., Offer, G. J., Hinds, G., Brett, D. J. L. & Shearing, P. R. (2015). *Nat. Commun.* **6**, 6924.
Garcia, C. & Delakis, M. (2004). *IEEE Trans. Pattern Anal. Mach. Intell.* **26**, 1408–1423.
Guizar-Sicairos, M., Thurman, S. & Fienup, J. (2008). *Opt. Lett.* **33**, 156–158.
Gürsoy, D., De Carlo, F., Xiao, X. & Jacobsen, C. (2014). *J. Synchrotron Rad.* **21**, 1188–1193.
Kanitpanyacharoen, W., Parkinson, D. Y., De Carlo, F., Marone, F., Stampanoni, M., Mokso, R., MacDowell, A. & Wenk, H.-R. (2013). *J. Synchrotron Rad.* **20**, 172–180.
Kingma, D. P. & Ba, J. (2014). *arXiv*:1412.6980.
Krizhevsky, A., Sutskever, I. & Hinton, G. E. (2012). *Advances in Neural Information Processing Systems 25*, edited by F. Pereira, C. J. C. Burges, L. Bottou and K. Q. Weinberger, pp. 1097–1105. Curran Associates, Inc.
Lawrence, S., Giles, C. L., Tsoi, A. C. & Back, A. D. (1997). *IEEE Trans. Neural Netw.* **8**, 98–113.
Lecun, Y., Bottou, L., Bengio, Y. & Haffner, P. (1998). *Proc. IEEE*, **86**, 2278–2324.
Mader, K., Marone, F., Hintermüller, C., Mikuljan, G., Isenegger, A. & Stampanoni, M. (2011). *J. Synchrotron Rad.* **18**, 117–124.
Matsugu, M., Mori, K., Mitari, Y. & Kaneda, Y. (2003). *Neural Netw.* **16**, 555–559.
Mirowski, P., Madhavan, D., LeCun, Y. & Kuzniecky, R. (2009). *Clin. Neurophysiol.* **120**, 1927–1940.
Moosmann, J., Ershov, A., Altapova, V., Baumbach, T., Prasad, M. S., LaBonne, C., Xiao, X., Kashef, J. & Hofmann, R. (2013). *Nature (London)*, **497**, 374–377.
Pereira, F., Mitchell, T. & Botvinick, M. (2009). *NeuroImage*, **45**, S199–S209.
Relch, E. S. (2013). *Nature (London)*, **501**, 148–149.
Vo, N. T., Drakopoulos, M., Atwood, R. C. & Reinhard, C. (2014). *Opt. Express*, **22**, 19078–19086.
Xiao, X., Fusseis, F. & De Carlo, F. (2012). *Proc. SPIE*, **8506**, 85060K.
Yang, Y., Yang, F., Hingerl, F. F., Xiao, X., Liu, Y., Wu, Z., Benson, S. M., Toney, M. F., Andrews, J. C. & Pianetta, P. (2015). *J. Synchrotron Rad.* **22**, 452–457.