Convolutional Neural Network For Single Particle Imaging With X-FEL

Haoyuan Li (hyli16@stanford.edu)
Xizhi Han (hanxzh@stanford.edu)

Introduction

Single particle imaging experiments produce diffraction patterns of microscopic particles (e.g. viruses and proteins), from which their 3D electron density can be reconstructed. An automatic binary classifier that selects high-quality samples is crucial for successful reconstruction.

The state-of-the-art method is diffusion map[3], a spectral clustering algorithm which groups similar diffraction patterns. However it is

- Computationally expensive for huge dataset,
- Incapable of yielding real time feedback for experiment diagnosis,
- Sensitive to quality of the dataset and its own hyperparameters,
- Only semi-automatic and requires manually labeling for each cluster.

In this project, two convolutional neural networks are developed as efficient and robust real-time classifiers of diffraction patterns.

Problem

The purpose is to classify diffraction patterns into two categories:

i) "Good", produced by incident X-ray diffraction from a single particle, showing concentric, symmetric and sharp diffraction fringes (see Fig. 1);

ii) "Bad", produced by diffraction from multiple particles or unwanted material, which is typically chaotic (see Fig. 2).

A ResNet[2] is trained on simulation and experimental data for image classification. A pretrained VGG16[1,6] is adopted for transfer learning.

Performance is evaluated by accuracy, speed and robustness against noises and changes of particle shape.

Dataset

Our dataset (before augmentation) contains 25275 experiment and 50000 simulation images, with 14770 and 25000 good ones respectively. Each image is 256 * 256 gray-scale.

Artifacts of this dataset include:

- Simulation data's fringe intensity is lower than that of experiment data;
- Simulation data is more symmetric and lacks large-scale fluctuations.

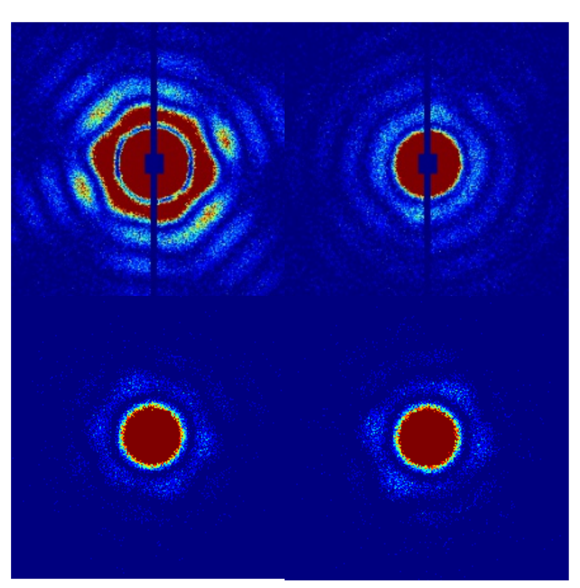

Figure 1. Examples of "good" images.
Up: real data. Down: simulation data.

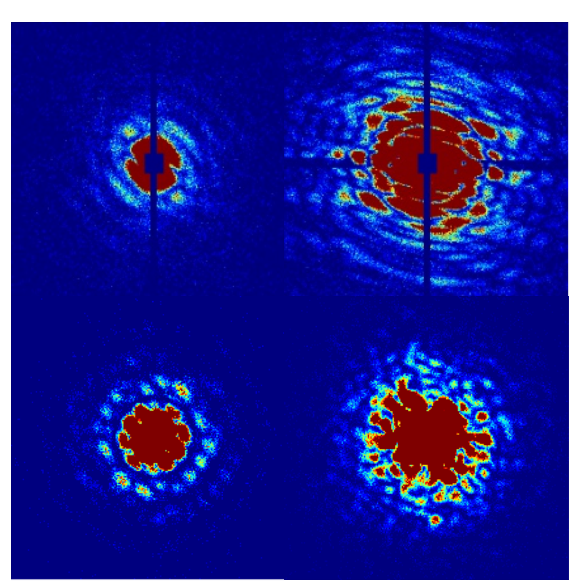

Figure 2. Examples of "bad" images.
Up: real data. Down: simulation data.

Methods

Data preprocessing and augmentation Due to lack of accurately labelled samples, we supplement real data with simulation data and further augment them by horizontal and vertical flipping. Each pixel is normalized to zero mean and unit variance across the training data to amplify invisible fringes (see Fig. 3) and images are cropped to 128 * 128 pixels near the center.

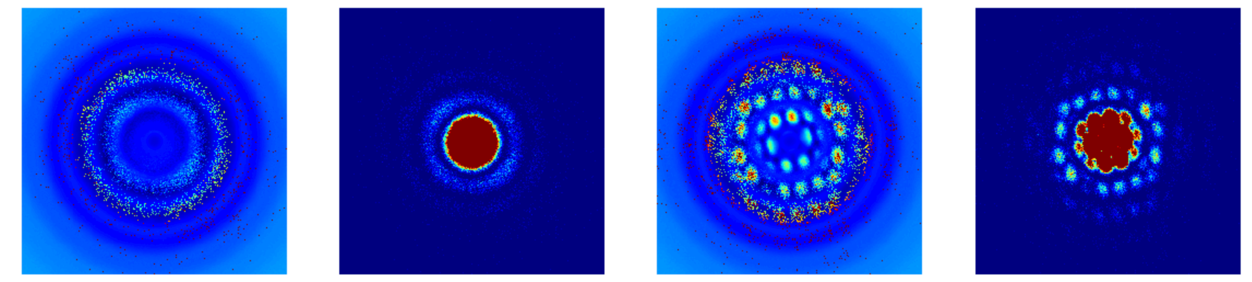

Figure 3. From left to right: 1, 3 are images corresponding to original data 2, 4, after renormalization.

ResNet Our architecture is adopted from [2], with a conv layer of output size 64 * 64 * 16 followed by a block of 2 n conv layers of output size 32 * 32 * 32 and then a block of 2 m conv layers of output size 16 * 16 * 64 (with identity map every two layers). Finally a 2 * 2 average pooling and then a fully connected layer give scores of two classes. Each conv layer is followed by a relu nonlinearity. Batch normalization and dropout are used.

VGG16 For comparison we extract 8192 feature maps in the last two layers of VGG16 trained on ImageNet and run a logistic regression from these features of training data to binary classification.

Evaluation

Accuracy First we present test accuracies of ResNet with n = 2, m = 2 on classification:

Training set / Test set	Sim / Sim	Exp / Exp	Sim / Exp
Test accuracy	99.8%	94.6%	62.0%

It is remarkable that the exp/exp accuracy is slightly lower than sim/sim accuracy, while sim/exp is unsatisfactory, which hints that our simulator needs to be improved.

The activations in different layers of ResNet and VGG16 are investigated and it seems that while VGG16 is looking for some edges, ResNet tends to process all dark or light areas as a whole. The shape of those regions is sensitive to large-scale fluctuation of intensity which has not been incorporated into our simulator.

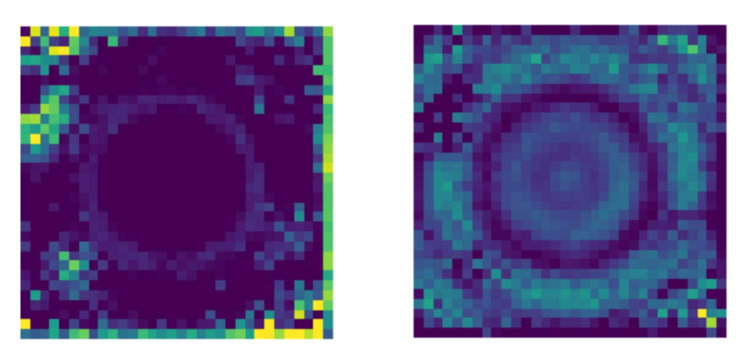

Figure 4. Activation maps in ResNet10 (cropped).

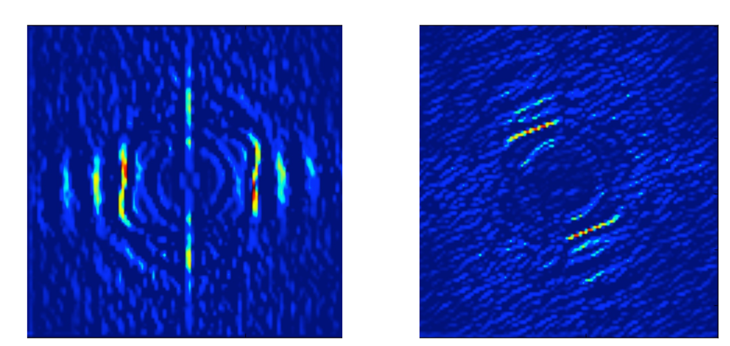

Figure 5. Activation maps in VGG16.

The exp/exp accuracy of VGG16 is 98.6%, which is better than RestNet with (n, m) = (2, 2). This may be a result of insufficient training data for ResNet, as increasing its depth leads to significant overfitting shown in Fig. 6.

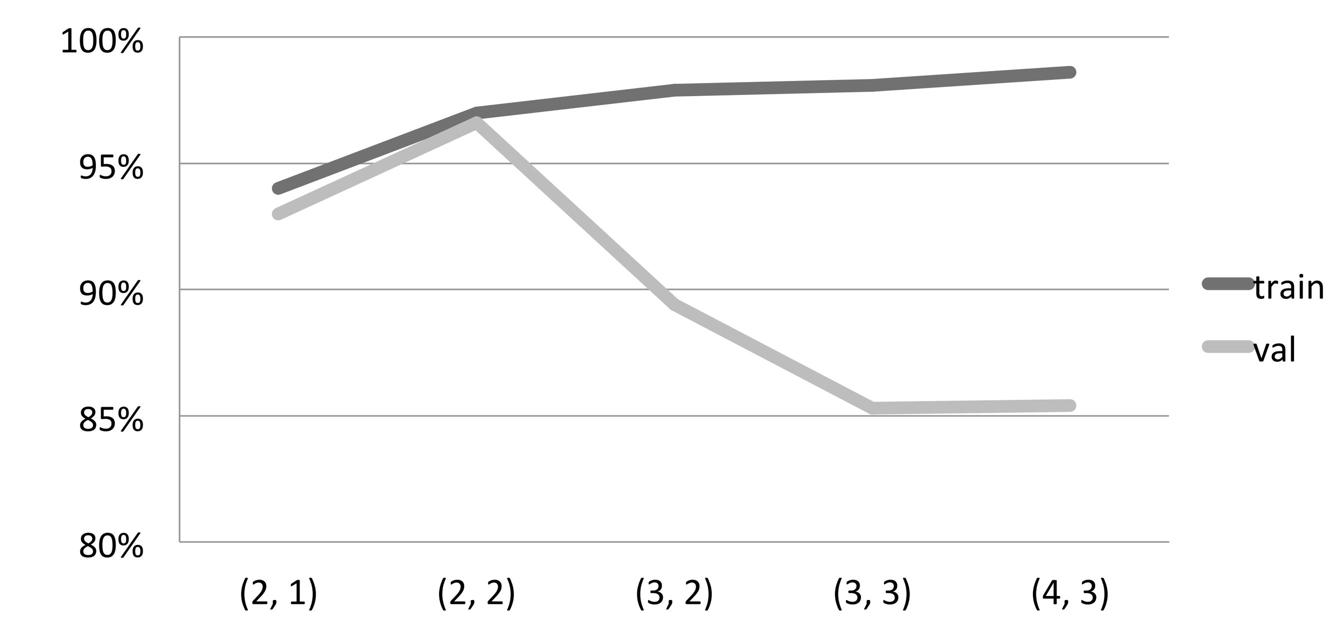

Figure 6. Training and validation accuracy for different number of conv layers (n, m) in ResNet.

Speed Diffusion map is able to process 7214 snapshots in 150 seconds on a desktop computer with a 2.66 GHz, 32 GB RAM, Quad-Core Intel Xeon CPU[3] and in comparison, our ResNet is able to give predictions for 7000 images in 30 seconds on similar hardware, which is a 5x speed-up with real-time capacity.

Robustness For real-world use our algorithm must work robustly for a variety of particles with distinct geometries. Our ResNet10 is trained on simulation data composed of half good and half bad images from diffraction of icosahedra, and tested on simulated diffraction from other types of particles (soccer and tori).

The result is as follows. Note that false positive rate (a bad image wrongly identified as good), which almost vanishes, deviates from false negative rate (a good image wrongly identified as bad), which is high.

Geometry	Icosahedron	Soccer	Torus
False Negative	0%	58.7%	84.8%
False Positive	0%	0%	0%

Although our program does not perfectly generalize to arbitrary species of particles, if it decides that one image is good, the image is good with high confidence. This would suffice in practice since for electron density reconstruction a low false positive rate is much more important than the overall accuracy.

Conclusions

In this project we have successfully applied convolutional neural network to the problem of single particle image classification. Our method is real-time, efficient and fully automatic and capable of replacing current mainstream diffusion map algorithm in many situations.

However, for new particle shapes that are not present in the training set, the false negative rate of our algorithm is high, which may be inefficient in practical use. And visualization of activations also indicates that our simulator is unable to produce data of experimental quality.

Our major challenge now is insufficiency of data. In the future we will focus on improving the simulator to enrich the dataset. Then it may be possible to deepen our ResNet and achieve better results.

References

1. Simonyan K, Zisserman A. Very deep convolutional networks for large-scale image recognition[J]. arXiv preprint arXiv:1409.1556, 2014.
2. He K, Zhang X, Ren S, et al. Deep residual learning for image recognition[C]//Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. 2016: 770-778.
3. Yoon C H, Schwander P, Abergel C, et al. Unsupervised classification of single-particle X-ray diffraction snapshots by spectral clustering[J]. Optics express, 2011, 19(17): 16542-16549.
4. Zhu Y, Ouyang Q, Mao Y. A deep convolutional neural network approach to single-particle recognition in cryo-electron microscopy[J].
5. Acciarri R, Adams C, An R, et al. Convolutional Neural Networks Applied to Neutrino Events in a Liquid Argon Time Projection Chamber[J]. arXiv preprint arXiv: 1611.05531, 2016.
6. For VGG16 code, see https://github.com/machrisaa/tensorflow-vgg.