Data Classification with Residual Network for Single Particle Imaging Experiments

Xizhi Han
Physics Department of Stanford University
Stanford CA
hanxzh@stanford.edu

Haoyuan Li
Physics Department of Stanford University
Stanford CA
hyli16@stanford.edu

Abstract

In this project, ResNet and VGG 16 are utilized as classifiers for dataset from single particle imaging experiments with X-ray free electron laser. Both ResNet and transfer learning with pre-trained VGG 16 achieve an accuracy over 94% on experiment dataset. Their performances are visualized and analyzed in detail.

Source code can be obtained from https://github.com/JustPhyCs/cs231n_proj

1. Introduction

One of the most persistent endeavor in molecular biology is to understand structure of various proteins essential to life. Historically structures of these tiny biological particles are determined by X-ray crystallography, leaving blanks in our knowledge of proteins that do not crystallize [11]. Recently a novel method called single particle imaging with X-ray free electron laser (SPI with XFEL), which does not require crystallization of samples, is revolutionarily shifting the landscape of structural biology [5, 24, 8].

The logic cornerstone of SPI with XFEL is as follows [18]: when the X-ray pulse is extremely short (i.e. with duration less than 10^{-15}s), diffraction patterns of the image are essentially slices of the particle in three-dimensional Fourier space; if furthermore the pulse is extremely strong (i.e. containing extremely many photons, for example 10^{14}), the signal-to-noise ratio is so high that reconstruction of electron density of the particle is possible.

Similar to other experiments running on accelerators, this new technology produces a huge amount of data. To reach a resolution of 0.5nm for a particle with a diameter of 100nm, more than 10^6 pieces of 2000×2000 pixel^2 diffraction images are required [10]. Due to certain physical constraints, only less than 0.1% experiment data are usable for reconstruction. Thus it is our goal to design a scalable, automatic, stable and precise binary classifier to select useful diffraction patterns from others.

In this work, we choose convolutional neural networks as our base model. A 10-layer ResNet [12, 3] is trained from scratch and VGG 16 [25, 3, 1] is adopted for transfer learning [22]. Both real experiment diffraction patterns and simulation data are utilized. The result shows that our approach is promising: transfer learning and ResNet yield an accuracy of 98% and 94% on real experiment data respectively.

2. History and Related Work

To the best of our knowledge we are the first group applying convolutional neural network (CNN) to the problem of SPI with XFEL.

At the beginning of SPI with XFEL, spectral clustering methods [28, 26, 13, 6, 21, 9, 4] and generative topological mapping methods [21, 7] competed on this classification problem. Generative topological mappings (GTM) are able to extract global orientation information, which is essential for reconstruction, as well as to classify. But the prohibitive amount of calculation resources renders it less useful among specialists.

For spectral clustering methods, diffusion maps are the most popular. They also extract the global orientation information, in principle. But the subtlety is that the extracted information is in "eigenfunction representation" (See Appendix A for detailed explanation). To incorporate information in this representation into reconstruction procedure after the classification, one has to know before-hand the shape of the underlying parameterizing manifold which we almost have no control over. But the undeniable advantage of these methods is that they are very easy to implement with highly optimized numerical linear algebra libraries [9].

Since GTM has not been applied to real experiment data yet, we only comment on the undesirable features of diffusion maps. Diffusion maps are unsupervised learning algorithms that do not require training and can be applied to any dataset. This is both an advantage and an disadvantage. On

one hand, without accurately labeled dataset these unsupervised learning algorithms are our only choice. In fact the real experiment data in this project is labeled by diffusion maps [28, 4].

On the other hand, they do not produce labels so researchers have to label each cluster manually. The number of clusters is not controlled explicitly by parameters of the algorithms and in one successful classification application, the researcher found more than 2000 clusters [28] to explore.

Also parameters for different datasets have to be tuned separately. Currently, there is no theoretical guideline for tuning but the performance sensitively depends on it. To our experience, the original diffusion map [9] is stable with respective to the parameters and the dataset quality (the ratio of good diffraction patterns among all data), but is not capable of accurately distinguishing good images from bad ones when the quality is bad. When the dataset quality is low, a lot of bad patterns are classified as good.

The normalized version of diffusion map [4], on the other hand, is very sensitive to parameters and the quality of the dataset, thus it is hard to tune parameters on bad datasets. Also, because such experiments have to rely on large X-ray facilities like LCLS, where the beam time is very expensive (for LCLS, the price is 2 US dollars per second), it would be very beneficial if one can know the quality of their dataset in real time and optimize their experiment parameters accordingly. Diffusion maps as global methods can not meet these requirements.

All these requirement can be satisfied by a well trained CNN. With CNN as is with human eye, each diffraction pattern can be classified properly. No additional orientation information is yielded. The processing time to scales only proportional to the number of the diffraction patterns and can be parallelized in a straightforward way. Also real time diagnosis of experiments and automatic labeling are available in this approach. This is the motivation of our project.

Supervised learning with CNN in XFEL are not very popular yet. Related research mainly concerns about X-ray diagnosis [23, 15]. On the other branch of single particle imaging, i.e. single particle imaging with cryogenic electron micro-scope (SPI with cryo-EM), there have been several attempts [29, 16, 27, 19, 20, 17]. But their dataset differs fundamentally from ours. SPI with cryo-EM reconstructs the 3D electron density from slices of the electron density while SPI with XFEL do the reconstruction with slices of the Fourier transformation of the electron density. So they need to find the positions of many particles on a huge image precisely, while we only need to decide whether the diffraction intensity patterns are good or not since the position of the particle has no effect on the intensity patterns.

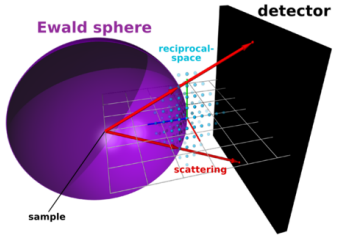

Figure 1. Relation between Fourier transformed electron density distribution and diffraction pattern recorded by detector. Image adopted from http://gisaxs.com/index.php/Ewald_sphere

3. Simulator and Dataset

Simulator is an indispensable part of our project. Our ultimate goal is to design a general classifier which can be applied to any dataset from SPI with XFEL. Thus the classifier should be able to distinguish good patterns from bad ones regardless the variations in X-ray intensity, detector configurations and particle shapes. But it is an intimidating task to find an accurately labeled representative experiment dataset for training. On one hand, there are not so many experiments yet in this field, and only very few different kinds of particles and configurations are explored. On the other hand, according to our experience, since the object is diffraction pattern, even human requires comprehensive training based on simulators to establish valid intuition for the classification. Unless one has seen similar diffraction patterns and knows the corresponding real space distributions, it is difficult to decide whether it's a good diffraction pattern or not.

Thus the first part of out project is to establish an acceptable simulator. Our simulator is only a toy model, merely considering the most relevant physics processes during diffraction. Many details are not taken into consideration. A more sophisticated model is now under construction by specialists in SLAC.

According to quantum mechanics, when the incident electromagnetic field is a plane wave with well defined wave vector k, the elastically scattered electromagnetic field obeys the formula up to an unimportant overall constant:

$$\tilde{f}(k_s) = \int d^3x e^{-ix\cdot(k_s-k_i)} f(x) \tag{1}$$

where k_s is the scattered wave vector and k_i is the incident wave vector.

Four fundamental geometric bodies are used as basic bricks in our simulator: octahedron, dodecahedron, icosahedron and truncated icosahedron (soccer ball shape). Each is represented by a 3-D ($128 \times 128 \times 128$) numpy array, with 1 representing that the point is inside the geometric body and 0 otherwise. Then combining simple geometric transformations (including twist, affine transformations),

the number of models scales up easily. With these bricks, it's straightforward to construct cluster with multiple geometric bodies. Four different clusters are constructed: clusters of two identical particles, three identical particles, four identical particles and more than four particles. As is indicated in the above formula, the Fourier transformations of the samples (recorded in a $256 \times 256 \times 256$ 3D numpy array) are taken and then the diffraction intensity is square of the magnitude.

As is shown in Fig. (1), there is a one-to-one correspondence between the detector pixel and the data point over the Ewald sphere. Thus, by randomly selecting an orientation and utilizing this relation, we get the diffraction intensity received by detectors. Those images are not the simulation final results. Due to quantum effect, photon are quantized and thus experiment data i.e. the photon numbers for each pixel must be integers. According to quantum optics, the photon number obeys Poisson distribution with an average equal to the intensity. Thus we generate a Poisson random number for each pixel which in the end constitutes our simulation data. To simulate the fluctuation of X-ray intensity, we multiply the real space models with different overall constants. Further to simulate the variation in detector configurations, we explore a large range of detector parameters in our simulator. To make our models more realistic, we use Gaussian filters to blur the edges.

In the end, we obtained four groups of simulated diffraction patterns corresponding to the four basic shapes. Each group has of 5.5×10^5 diffraction patterns with 2.5×10^5 of them being good patterns. Every 100 diffraction patterns comes from an independently initialized model (randomly generated orientations, X-ray intensities, geometric transformations, and detector configurations) to prevent over sampling.

Besides simulation patterns, 2.4×10^5 real experiment diffraction patterns are also utilized [28, 4]. 1.4×10^5 of them are good hits. These data are labeled by the diffusion maps mentioned above. The precise accuracy of the labeling is unknown. According to inspection by the author over 500 randomly chosen sample, the accuracy may be around $95\% - 98\%$. But since we know the accuracy of this data set is good enough for successful reconstruction, thus so long as our classifier's predictions compatible with diffusion maps, it is good enough for our purpose.

To gain more intuition of our dataset, some samples are shown in Fig. (2). Diffraction patterns from different number of particles hit by X-ray are visually distinct: diffraction of more particles shows more low-intensity stripes due to destructive interference between particles. However these stripes can be distorted, rotated, rescaled or blurred because of noise or different spatial configurations of particle clusters.

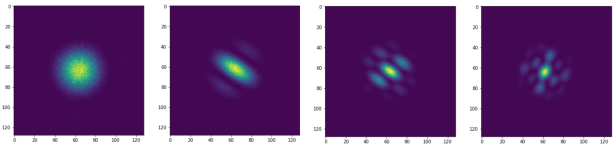

Figure 2. Examples of simulation data (diffraction intensity from icosahedron particles). From left to right: single, double, triple and four hit.

4. Methods

In this work, we propose to use convolutional neural networks to recognize subtle interference patterns in diffraction images. Concerned about insufficient data, we implement a 10-layer ResNet trained on simulation and experiment data, and a logistic classifier based on pre-trained VGG16 over ImageNet for comparison.

4.1. ResNet

A ResNet [12, 3, 2] classifier that decides the number of particles (from zero to four) hit by X-ray in each snapshot is trained and tested on simulation and experiment data. The optimized loss function is

$$L = -\frac{1}{N}\sum_{i=1}^{N}\log\frac{s_{i,y_i}}{\sum_{j=1}^{C} s_{i,j}} + \lambda\sum_{k}\|W_k\|^2, \qquad (2)$$

where N is the number of samples in the batch, $s_{i,j}$ is the score for sample i in class j output by the fully connected (fc) layer, y_i is the correct class for sample i, and $C = 5$ is the total number of classes. $\lambda = 0.01$ is the L^2 regularization parameter for convolutional kernels W_k.

The network architecture is similar to that in [12] with identity mapping by shortcuts for every two convolutional layers, as is shown in Fig. (3). Since we are only classifying images into five classes a 10-layer network is sufficient in practice. Each convolutional filter recognizes 3×3 activation patterns of previous neurons around each pixel in image (for example the first convolutional layer seems to detect edges); shortcut connections act as highways of gradients in back-propagation and hence make training of the network more efficient.

We have also preprocessed and augmented data for ResNet. Note that average intensity of pixels is much larger in the middle so for numerical stability, we renormalize the image to zero mean value and unit variance across all training data. This helps amplify stripe patterns in images as well (see Fig. (4)). Also to improve efficiency we only keep 128×128 pixels in the center of the image for classification. Furthermore images are horizontally and vertically flipped to enlarge the dataset.

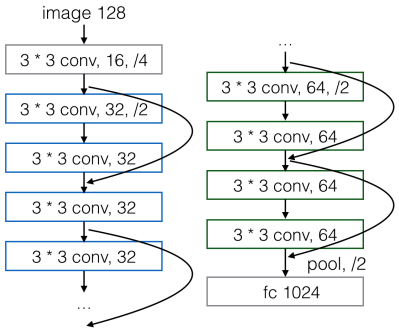

Figure 3. Network architecture. Each convolutional layer is appended with a batch normalization, a dropout and then a relu layer.

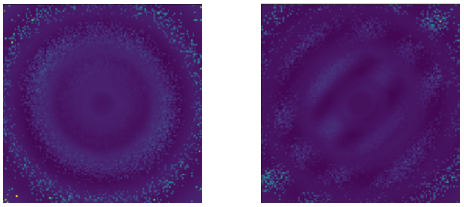

Figure 4. Single hit (left) and double hit (right) with mean image subtracted and normalized to unit variance. The interference stripes are more evident than that in Fig. (2).

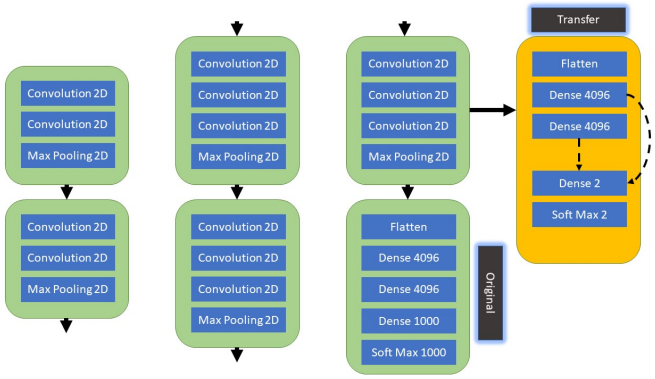

Figure 5. Transfer learning network architecture. The yellow block represents the adopted structure.

4.2. VGG 16

In our project, VGG 16 [1, 22, 25, 3] is utilized as the base model for transfer learning. The transfer learning is based on source code on github https://github.com/machrisaa/tensorflow-vgg. As is shown in Fig. (5), we keep all convolutional layers and the first two fully connected layers frozen, use the output from the two fully connected layers as features of the diffraction patterns and train a logistic regression classifier on the $8192D$ features.

One of the basic guidelines for transfer learning is to use pre-trained model over similar training set. This can guarantee that useful features are extracted. Since the diffraction patterns are significantly different from the training data of VGG 16 on ImageNet. Special justification and preprocessing are required for this approach.

First we inspect the diffraction patterns at different intensity scales, as is shown in Fig. (6) and Fig. (7).

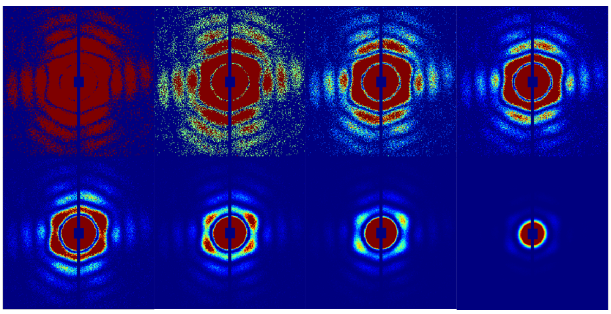

Figure 6. A typical good diffraction pattern viewed at different intensity scales. From left to right, from top to bottom, the highest intensity per pixel are respectively 1,2,5,10,20,50,100,1000

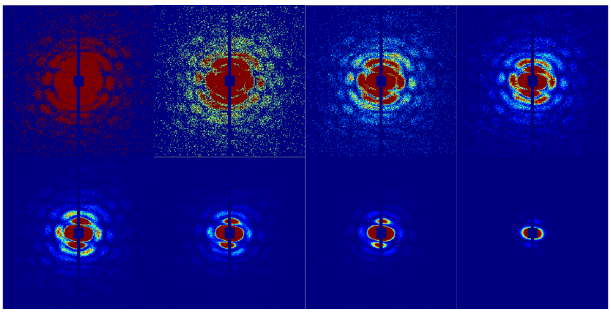

Figure 7. A typical bad diffraction pattern viewed at different intensity scales. From left to right, from top to bottom, the highest intensity per pixel are respectively 1,2,5,10,20,50,100,1000

Obviously, at different scale, different features emerge. It seems to us that scaling being 10 keeps enough features of the patterns while suppressing the noise to a decent degree. So we save the diffraction patterns with maximal intensity scale being 10 in .jpg format. In this way, we also finish the normalization procedure.

A lot of information is lost during the format transformation. But we can still justify the procedure. The only desired information in our project is whether the diffraction pattern is good or not. This transformation doesn't seem to destroy this information. Also through this transformation, we obtain an analogy between diffraction pattern and training samples in ImageNet. We can deem the diffraction patterns as a "special family of flowers". Some of the kinds have regular petals while the others have irregular ones. Since VGG 16 can successfully distinguish much more complicated categories, there is no doubt that fine tuned VGG 16 can meet the requirement.

5. Results and Discussion

5.1. ResNet

As a preliminary experiment we train the network over 58000 simulation images and reserve 1000 images for validation and 1000 for test. These 60000 images contain 30000 single hit images of icosahedron and soccer-shaped particles, 10000 double hit, 10000 triple hit, 5000 four hit and 5000 bad hit images. The loss, training accuracy and validation accuracy profiles are plotted in Fig. (8). The accuracy quickly gets over 0.95 and there is no significant overfitting (although there is a dip in validation accuracy at the beginning of training). A final test accuracy as high as 99% seems to suggest that such classification task of simulation data is too easy for a deep neural network.

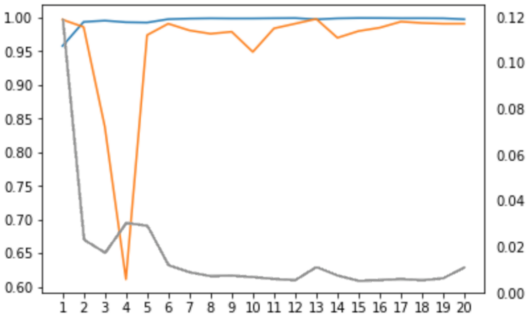

Figure 8. Loss averaged over each epoch is drawn as the gray line with values on the right. The blue line is training accuracy and the orange line is validation accuracy, averaged over each of 20 epochs of training over 58000 simulation images. A 25 percent dropout is adopted with learning rate 1e-3 and regularization 0.01.

Nevertheless, to better understand the performance of ResNet, it should be helpful to analyze some unsuccessful classification results. In Fig. (9) we present some misclassified images and compare them to successful ones. It should be noted that the more particles hit by X-ray the more difficult it is to determine the number of particles involved because then noise would destroy details of the intricate interference pattern. Hence images with higher resolution may be needed in order to recover such details.

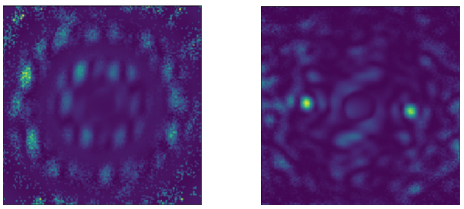

Figure 9. Left: a four hit images wrongly classified as bad hit; right: a bad hit correctly identified. When many particles are hit by X-ray the interference pattern gets complicated and easier to be misclassified as bad hit.

However, when the ResNet trained on simulation data is tested on experiment images, its accuracy drops to 62.0%. First we observe that this is not the result of incapacity of ResNet, as for the same network trained and tested both on experiment data, it reaches an accuracy of 94%, and increase of network depth leads to significant overfitting as is shown in Fig. (10).

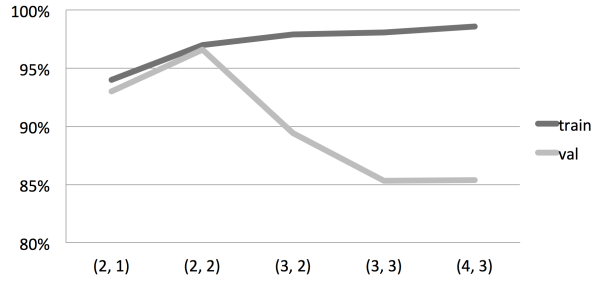

Figure 10. Training and validation accuracy for different number of convolutional layers in ResNet. (n, m) are the numbers of convolutional layers with 32 and 64 channels respectively.

Although the overall test accuracy on experiment snapshots is only 62%, the false positive (non-single hit mistaken as single hit) rate nearly vanishes. Hence the problem is our ResNet trained on simulation data tends to classify single-hit images in experiment data as multiple-hit. By studying misclassified examples, we conclude that such failure of transferring from simulation to experiment data may be a result of several artifacts in our simulation dataset. From Fig. (11) we observe that simulation data is more symmetric and lacks large-scale fluctuations. Also intensity of fringes in simulation data is visibly lower than that in real images. These discrepancies may fool our ResNet to regard less clean experiment single-hit snapshots as multiple-hit.

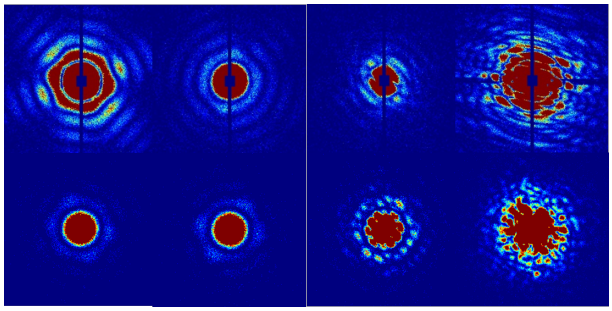

Figure 11. Examples of experiment diffraction patterns (top) and simulation patterns (bottom).

Another concern is robustness of our network to particles not present in training set. In our simulation data for training, the particles are highly symmetric (icosahedra and soccer) but in real experiments particles can take a variety of geometries. It should be good to test the model over some new particles. In such an experiment our ResNet 10 is trained on simulation data composed of half single-hit

and half multiple-hit 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 a clean electron density reconstruction a low false positive rate is more important than the overall accuracy.

5.2. VGG 16

We use scikit-learn package [22, 3, 1] to implement the logistic classifier since the data set is small. Only experiment data is investigated. The default parameters are utilized and the regularization strength for L^2 penalty is set to 1. The training is carried out on 2×10^5 randomly chosen real experiment samples, while the test is over 2000 randomly chosen real experiment samples. The final test accuracy is 98.2%. Considering that the accuracy of the ground truth is not this high, it seems to us that to pursue an even higher accuracy in current situation is meaningless.

Prediction statistics over 2000 randomly chosen test data is listed below. There is no obvious bias for either kind of mis-prediction.

Truth/Prediction	Good/Good	Bad/Bad	Good/Bad	Bad/Good
Number	1166	798	13	23

We inspect all of the misclassified samples. The ground truth for these samples are correct and thus these are true mis-classifications. For good diffraction patterns misclassified as bad ones, it seems that X-ray intensity is a major effect. Because most of the misclassified good diffraction patterns have low X-ray fluxes as is shown in the first image in (12). For our specific purpose, this is actually beneficial. Since too many diffraction patterns with low X-ray fluxes can potentially impede the next reconstruction procedure. As for bad diffraction patterns, we do not see any particular reason leading to the mis-classification.

To get a hint of what the pretrained neural network is looking at, we inspect the output of middle layers. It seems that at the first several layers, the neurons mainly looks for major color blocks while at higher layers, neurons tends to look for edges.

Since the precision is very high, it seems reasonable to assume that the extracted feature is highly separable. So we inspect the 2 component t-SNE mapping over the extracted

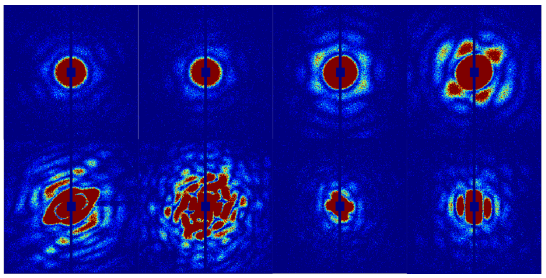

Figure 12. Up:Good diffraction patterns predicted to be bad. Down: Bad diffraction patterns predicted to be good

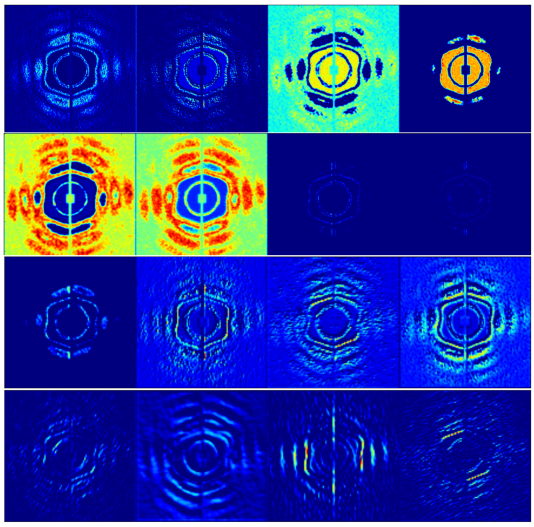

Figure 13. Selected output from neurons at various layers. From top to bottom, neurons respectively are located at 1st, 2nd, 3rd and 4th convolution layer.

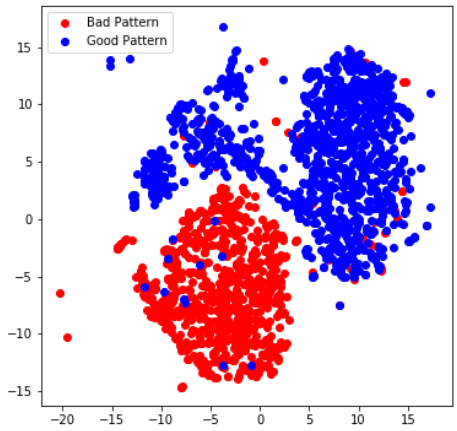

Figure 14. t-SNE embedding of 2000 $8192D$ test sample features

8192D features of the 2000 test samples [22]. Clearly, the embedding shows a very good separation (See Fig.(14)).

This indicates that our normalization methods is well suited for transfer learning with VGG 16.

6. Conclusion and Future Work

In this project, binary classifiers implemented with ResNet and VGG 16 are constructed for dataset from SPI with XFEL. A binary classification accuracy as high as 98% of VGG 16 is suffcient for immediate application in processing data from XFEL diffraction experiments. Transfer learning of VGG 16 shows advantages of higher accuracy with limited dataset. However we believe that with our simulator improved and dataset enlarged, we would be able to deepen our ResNet and achieve higher accuracy and robustness, as ResNet is specially trained on diffraction images and it has already shown advantages in other image classification problems.

In the future, one major concentration should be to improve the simulator. With better simulator, it is possible to train a better ResNet from scratch and improve its generalization ability. Another concentration should be fine tuning VGG 16 or doing transfer learning with pretrained ResNet. Besides, improving the labeling condition in the training set is an urgent task now, since currently, the accuracy of the labeling in real experiment data is not high enough to correctly reflect the classifiers' performance.

For this project, source code can be obtained from https://github.com/JustPhyCs/cs231n_proj

7. Appendix A

By eigenfunction representation, we refers to a special way of representing data points over the data manifold.

Supposing that our data points are images with pixel values representing the coordinates, then initially our data points are embedded in a very high dimensional space. But we may know or assume that the data points are parameterized by only a few parameters and only occupy a low dimensional manifold of the high-dimensional space.For simplicity, we assume that the low dimensional manifold is a 2D torus T^2.

Then with GTM[7, 21], the extracted information would be the positions of the data points over the torus under certain parameterization of the torus. With diffusion maps [9], the extracted information would be very different. Supposing that ϕ_n is the nth non-trivial eigenfunction of Laplacian operator over the torus with the nth lowest eigenvalues, then the nth component of the extracted information is the value of the nth eigenfunction over the specific data point. Mathematically, it can be proved that for 2-D manifold [14], only when the genus of the manifold is 0 can we infer the topological structure of the manifold uniquely. But there's only one such 2-D topological manifold in the world, S^2. Thus unless we know the shape of the manifold before-hand, we can not interpret the extracted information.

References

[1] https://github.com/machrisaa/tensorflow-vgg. Accessed May 29, 2017. 1, 4, 6

[2] Assignments code of cs231n class, http://cs231n.github.io/. 3

[3] M. Abadi, A. Agarwal, P. Barham, E. Brevdo, Z. Chen, C. Citro, G. S. Corrado, A. Davis, J. Dean, M. Devin, S. Ghemawat, I. Goodfellow, A. Harp, G. Irving, M. Isard, Y. Jia, R. Jozefowicz, L. Kaiser, M. Kudlur, J. Levenberg, D. Mané, R. Monga, S. Moore, D. Murray, C. Olah, M. Schuster, J. Shlens, B. Steiner, I. Sutskever, K. Talwar, P. Tucker, V. Vanhoucke, V. Vasudevan, F. Viégas, O. Vinyals, P. Warden, M. Wattenberg, M. Wicke, Y. Yu, and X. Zheng. TensorFlow: Large-scale machine learning on heterogeneous systems, 2015. Software available from tensorflow.org. 1, 3, 4, 6

[4] A. Aquila. Coherent soft x-ray diffraction imaging of coliphage pr772 at the linac coherent light source. Technical report, SLAC National Accelerator Laboratory (SLAC), 2017. 1, 2, 3

[5] A. Aquila, A. Barty, C. Bostedt, S. Boutet, G. Carini, D. DePonte, P. Drell, S. Doniach, K. Downing, T. Earnest, et al. The linac coherent light source single particle imaging road map. *Structural Dynamics*, 2(4):041701, 2015. 1

[6] M. Balasubramanian and E. L. Schwartz. The isomap algorithm and topological stability. *Science*, 295(5552):7–7, 2002. 1

[7] C. M. Bishop, M. Svensén, and C. K. Williams. Gtm: The generative topographic mapping. *Neural computation*, 10(1):215–234, 1998. 1, 7

[8] H. N. Chapman, A. Barty, M. J. Bogan, S. Boutet, M. Frank, S. P. Hau-Riege, S. Marchesini, B. W. Woods, S. Bajt, W. H. Benner, et al. Femtosecond diffractive imaging with a soft-x-ray free-electron laser. *Nature Physics*, 2(12):839–843, 2006. 1

[9] R. R. Coifman and S. Lafon. Diffusion maps. *Applied and computational harmonic analysis*, 21(1):5–30, 2006. 1, 2, 7

[10] J. Flamant, N. Le Bihan, A. V. Martin, and J. H. Manton. Expansion-maximization-compression algorithm with spherical harmonics for single particle imaging with x-ray lasers. *Phys. Rev. E*, 93:053302, May 2016. 1

[11] S. Hau-Riege. Toward single-particle bioimaging using x-ray free-electron lasers. Technical report,

Lawrence Livermore National Laboratory (LLNL), Livermore, CA, 2013. 1

[12] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 770–778, 2016. 1, 3

[13] S. Kassemeyer, J. Steinbrener, L. Lomb, E. Hartmann, A. Aquila, A. Barty, A. V. Martin, C. Y. Hampton, S. Bajt, M. Barthelmess, et al. Femtosecond free-electron laser x-ray diffraction data sets for algorithm development. *Optics express*, 20(4):4149–4158, 2012. 1

[14] O. Lablée. *Spectral Theory in Riemannian Geometry*. 2015. 7

[15] H. Li and Q. Yin. Beam detection based on machine learning algorithms. *Statistics and computing*, 14(3):199–222, 2004. 2

[16] R. Li, D. Si, T. Zeng, S. Ji, and J. He. Deep convolutional neural networks for detecting secondary structures in protein density maps from cryo-electron microscopy. In *Bioinformatics and Biomedicine (BIBM), 2016 IEEE International Conference on*, pages 41–46. IEEE, 2016. 2

[17] R. Li, D. Si, T. Zeng, S. Ji, and J. He. Deep convolutional neural networks for detecting secondary structures in protein density maps from cryo-electron microscopy. In *Bioinformatics and Biomedicine (BIBM), 2016 IEEE International Conference on*, pages 41–46. IEEE, 2016. 2

[18] S. Marchesini, H. Chapman, S. Hau-Riege, R. London, A. Szoke, H. He, M. Howells, H. Padmore, R. Rosen, J. Spence, et al. Coherent x-ray diffractive imaging: applications and limitations. *Optics Express*, 11(19):2344–2353, 2003. 1

[19] J. V. Mascarenhas, C. Yang, and D. Ushizima. Quantization for energy efficient convolutional neural networks. 2

[20] W. V. Nicholson and R. M. Glaeser. Review: automatic particle detection in electron microscopy. *Journal of structural biology*, 133(2-3):90–101, 2001. 2

[21] A. Ourmazd, P. Schwander, and G. N. Phillips Jr. Structure from random snapshots. In *SPIE Optical Engineering+ Applications*, pages 780002–780002. International Society for Optics and Photonics, 2010. 1, 7

[22] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830, 2011. 1, 4, 6

[23] A. Sanchez-Gonzalez, P. Micaelli, C. Olivier, T. Barillot, M. Ilchen, A. Lutman, A. Marinelli, T. Maxwell, A. Achner, M. Agåker, et al. Accurate prediction of x-ray pulse properties from a free-electron laser using machine learning. *Nature communications*, 8:15461, 2017. 2

[24] M. M. Seibert, T. Ekeberg, F. R. Maia, M. Svenda, J. Andreasson, O. Jönsson, D. Odić, B. Iwan, A. Rocker, D. Westphal, et al. Single mimivirus particles intercepted and imaged with an x-ray laser. *Nature*, 470(7332):78–81, 2011. 1

[25] K. Simonyan and A. Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014. 1, 4

[26] Y. Ueno, M. Kawata, and S. Umeyama. Intrinsic classification of single particle images by spectral clustering. *Proceedings of Biosignal Processing and Classification. Portugal: INSTICC Press*, pages 60–67, 2005. 1

[27] F. Wang, H. Gong, G. Liu, M. Li, C. Yan, T. Xia, X. Li, and J. Zeng. Deeppicker: A deep learning approach for fully automated particle picking in cryo-em. *Journal of Structural Biology*, 195(3):325–336, 2016. 2

[28] C. H. Yoon, P. Schwander, C. Abergel, I. Andersson, J. Andreasson, A. Aquila, S. Bajt, M. Barthelmess, A. Barty, M. J. Bogan, et al. Unsupervised classification of single-particle x-ray diffraction snapshots by spectral clustering. *Optics express*, 19(17):16542–16549, 2011. 1, 2, 3

[29] Y. Zhu, Q. Ouyang, and Y. Mao. A deep convolutional neural network approach to single-particle recognition in cryo-electron microscopy. 2