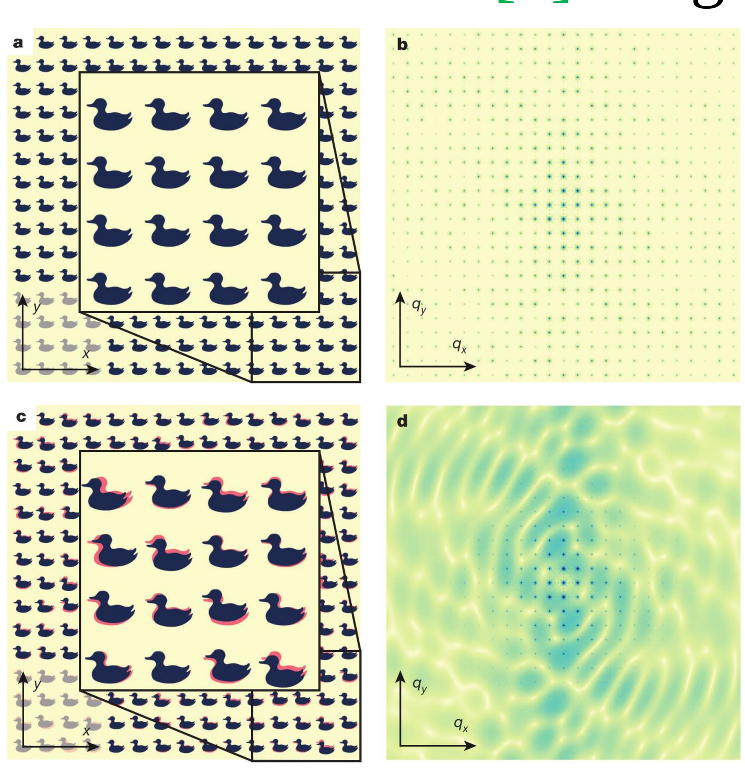
Classification of single-crystal diffuse scattering images

Zhen Su （Team ID: 104）
Department of Applied Physics, Stanford University
zhensu@stanford.edu

Abstract

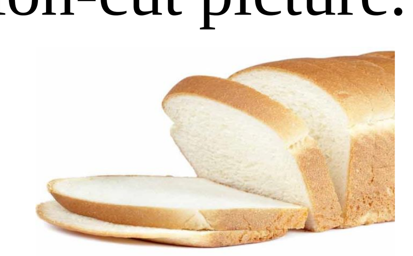
Every image collected from x-ray diffraction experiments is a cross section of the 3D Fourier transformation of the measured object. Among thousands of "good" images, there are also many different categories, where one important type is the diffuse scattering image [1]. My project is to extract those special images with tiny diffuse signals. To achieve this goal, I tried several machine learning algorithms including the diffusion map method, logistic regression and the CNN algorithm based on VGG16 [2] weight matrices. The classification accuracy is about 92% with reasonable preprocesses.



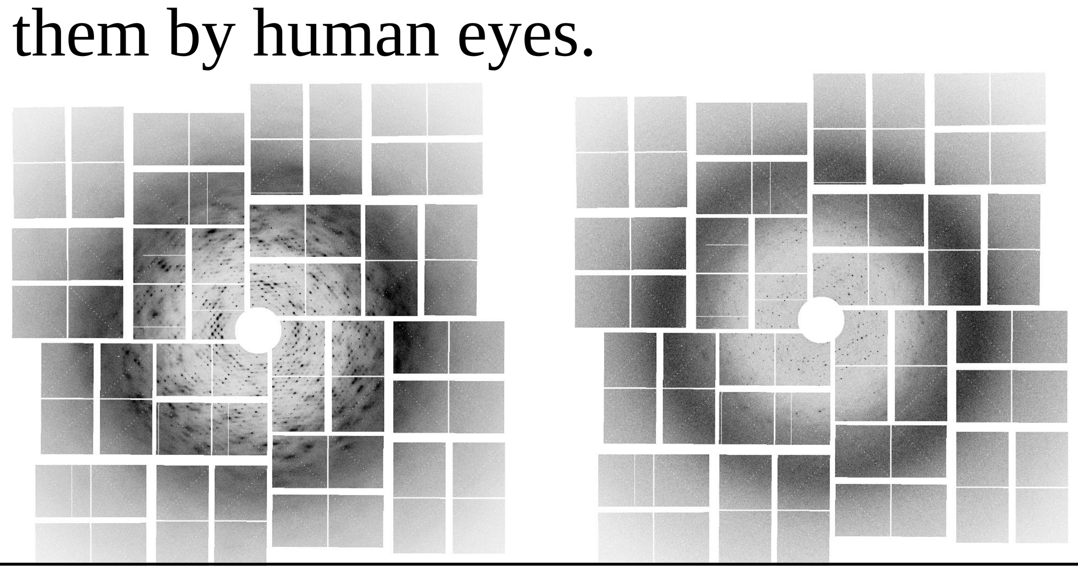
Images from [1]

Background

An analogy to my project is to classify two types of breads given the section-cut picture.



However, the difference between "non-diffuse" and "diffuse" images is much smaller compared to the background, sometimes it's even hard to label them by human eyes.



Diffusion Map

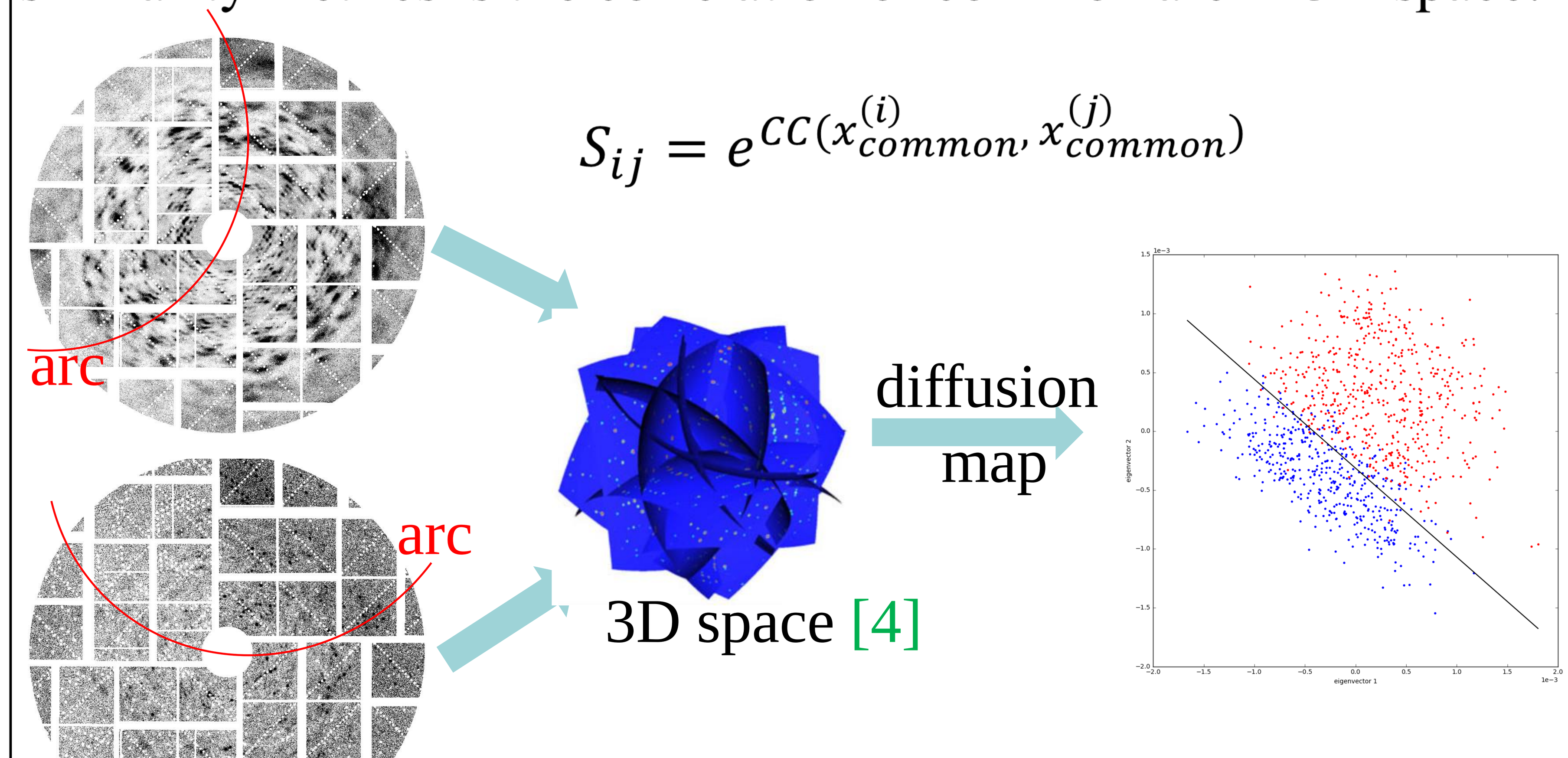
Model 1: Input vector $x^{(i)} \in \mathbb{R}^{1734} \times \mathbb{R}^{1731}$ is the normalized raw image or preprocessed image (remove the local or isotropic background). The distance D_{ij} or similarity matrix S_{ij} is:

$$D_{ij} = \left\| x^{(i)} - x^{(j)} \right\|^2 \quad \text{(Euclidean distance)}$$

$$S_{ij} = e^{CC(x^{(i)},\, x^{(j)})} \quad \text{(Correlation Coefficient)}$$

They failed to classify the "non-diffuse" and "diffuse", due to the sporadic distribution of diffuse features.

Model 2: I merged every image back to the 3D Fourier space, so each input vector became: $x_{new}^{(i)} \in \mathbb{R}^{200} \times \mathbb{R}^{200} \times \mathbb{R}^{200}$, The similarity metrics is the correlation of common arc in 3D space:

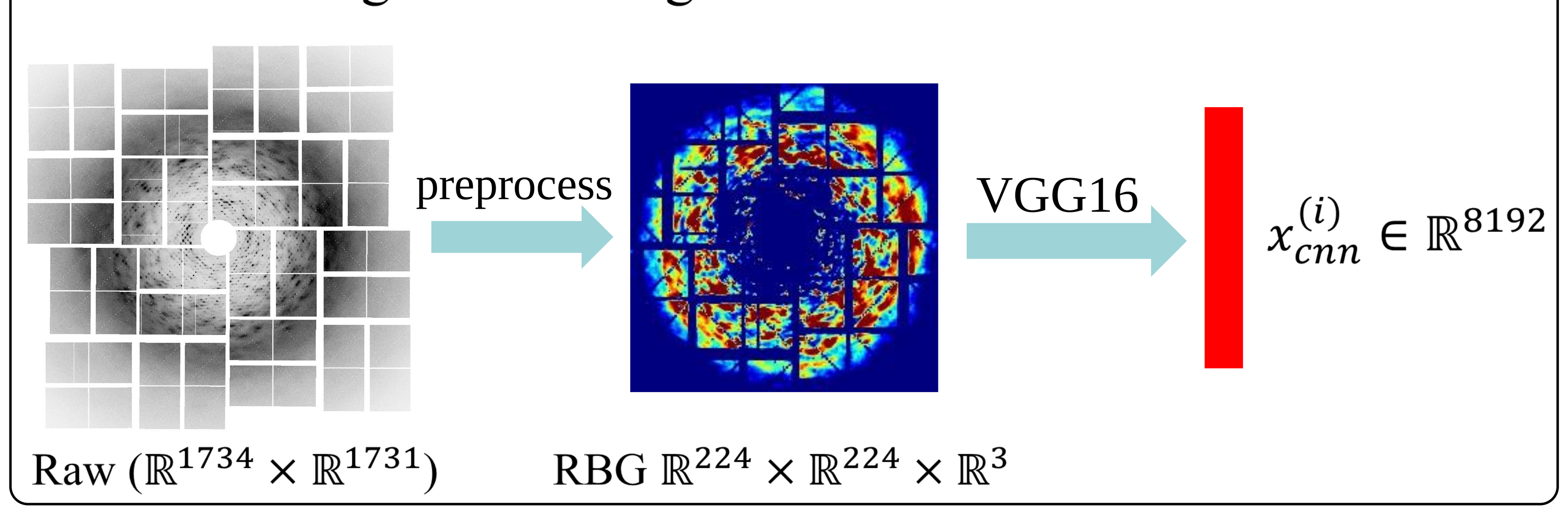


High accuracy (91.3%) but slow

Convolutional NN

VGG16 is a famous 16-layer convolutional neural network model [3]. We ran VGG16 with public weight matrices and use the output $x_{cnn}^{(i)} \in \mathbb{R}^{8192}$ for further logistic regression.

Input vector for VGG16 is rbg format $x^{(i)} \in \mathbb{R}^{224} \times \mathbb{R}^{224} \times \mathbb{R}^{3}$ of the raw image after background removal and normalization.



Logistic Regression

Each input for logistic regression is $x^{(i)} \in \mathbb{R}^{50}$, which is the principle components analysis of VGG16 output $x_{cnn}^{(i)} \in \mathbb{R}^{8192}$. I used 500 images for training and 500 images for testing. The overall test accuracy is about 92%. In detail, the "non-diffuse" image has higher recall rate, while the "diffuse" image has higher "precision" rate.

Classes	Precision rate	Recall rate
"diffuse"	95%	90%
"non-diffuse"	89%	95%
overall	92%	92%

Future Work

- Reduce or standardize the preprocess step
- Classify diffraction images to more than 2 categories with different amount of diffuse signals
- Test whether it can detect "diffuse" images from another experiment. If it doesn't work, then it means that we have to label plenty of images every time we do an experiment

References

[1] Ayyer, Kartik, et al. Nature 530.7589 (2016): 202-206.
[2] Simonyan, Karen, et al. arXiv:1409.1556 (2014).
[3] https://github.com/machrisaa/tensorflow-vgg
[4] Yefanov, Oleksandr, et al. Phil. Trans. R. Soc. B 369.1647 (2014): 20130333.

Acknowledge

Thanks to Dr. Chun Hong Yoon and Haoyuan Li for the valuable discussions and suggestions. Diffraction images were collected at SLAC National Laboratory by LP95 team in a recent experiment "cxiclp9515" for studying the DNA repair reaction led by Thomas J Lane (tjlane@slac.stanford.edu)