Explainable Machine Learning for Diffraction Patterns

Shah Nawaz[1], Vahid Rahmani[1], Shabarish Pala Ramakantha Setty[1], David Pennicard[1], and Heinz Graafsma [1,2]

[1]Deutsches Elektronen-Synchrotron DESY, Germany
[2]Mid-Sweden University, Sundsvall, Sweden

1 Abstract

Serial crystallography experiments at X-ray Free-Electron Laser facilities produce massive amounts of data. However, only a fraction of the data is useful for downstream analysis. Thus, it is essential to differentiate between acceptable and unacceptable data, generally known as 'hit' and 'miss' respectively. To this end image classification methods from artificial intelligence, or more specifically convolutional neural networks, classify the data into 'hit' and 'miss' categories in order to achieve data reduction. The quantitative performance from previous work indicates that convolutional neural networks successfully classify serial crystallography data into desired useful categories [9]. However, it does not provide qualitative evidence on internal workings of these networks while classifying serial crystallography data. For example, there is no visualization method that highlights features contributing to a specific prediction. Therefore, existing convolutional neural networks classifying serial cystallograhy data are like a 'black box'. To this end, we present a qualitative study to unpack internal workings of convolutional neural networks with an aim to visualize information in fundamental blocks of a standard network with serial crystallography data. In addition, we visualize region(s) or part(s) of an image that mostly contribute to a 'hit' or 'miss' prediction.

2 Introduction

Serial femtosecond crystallography (SFX) has become popular in determining biological structure from crystal diffraction patterns using X-ray Free Electron Laser (XFEL) sources [4, 23]. Using X-ray pulses, experiments can produce strong patterns from weakly diffracting crystals

at room temperature. However, these pulses also destroy the crystals, so diffraction patterns need to be gathered from many crystals. This results in large quantities of data, for example, The Coherent X-ray Imaging instrument at the Linac Coherent Light Source (LCLS) delivers full frames of data at up to 120 Hz, producing 432,000 samples per hour and data of tens to hundreds of terabytes in size [1]. In spite of the large data volumes produced, only a small percentage of the data is useful for downstream analysis. Figure 1 illustrates a typical experimental setup at the European X-ray Free-Electron Laser (Eu.XFEL). Protein crystals are fired through the path of the X-ray beam in a liquid jet, and only a small proportion of X-ray pulses will actually hit a crystal. For example, an early SFX experiment at Eu.XFEL involving CTX-M-14 β-lactamase produced 3,215,616 images at an average rate of 300 images per second. Of these, only 14,445 images (0.4%) were observed to contain diffraction patterns from protein crystals as observed in the offline analysis [23]. In this early experiment, both the accelerator and detector were operated below their maximum rates; at full speed, up to 3520 images per second can be taken. Furthermore, new free-electron laser facilities such as LCLS-II will handle experiments with continous repetition rates up to 1 MHz, resulting in even higher data volumes [6].

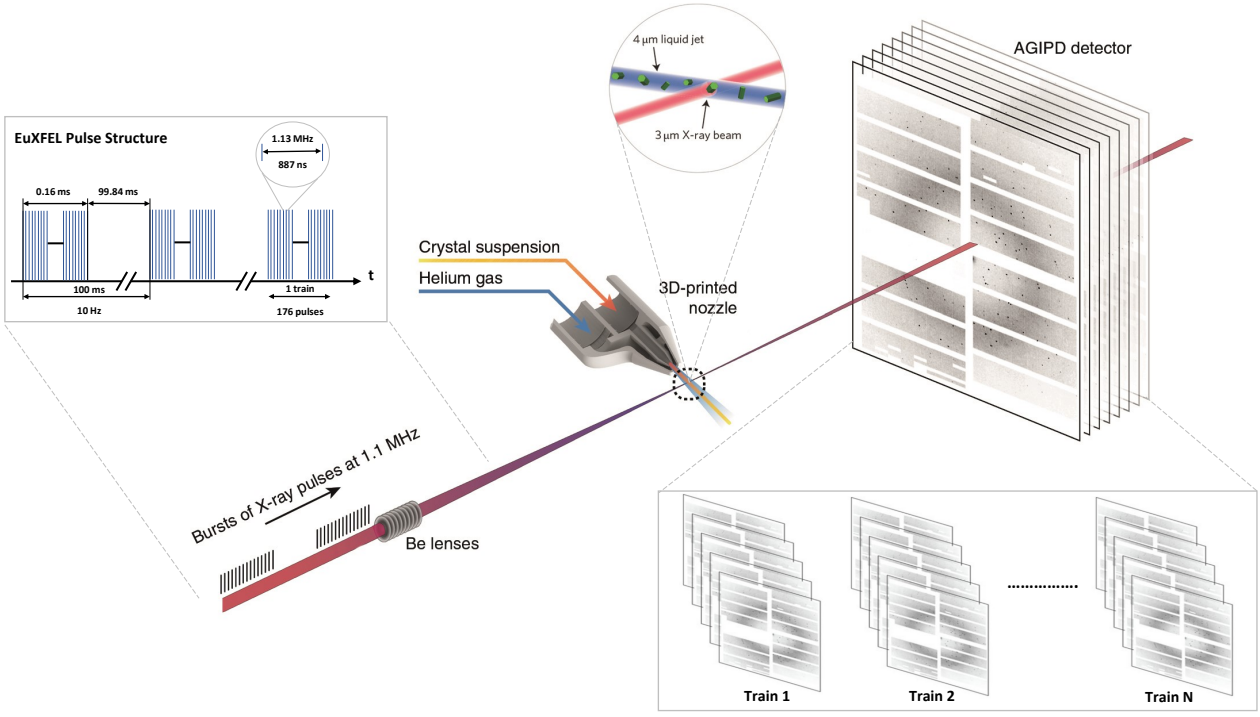
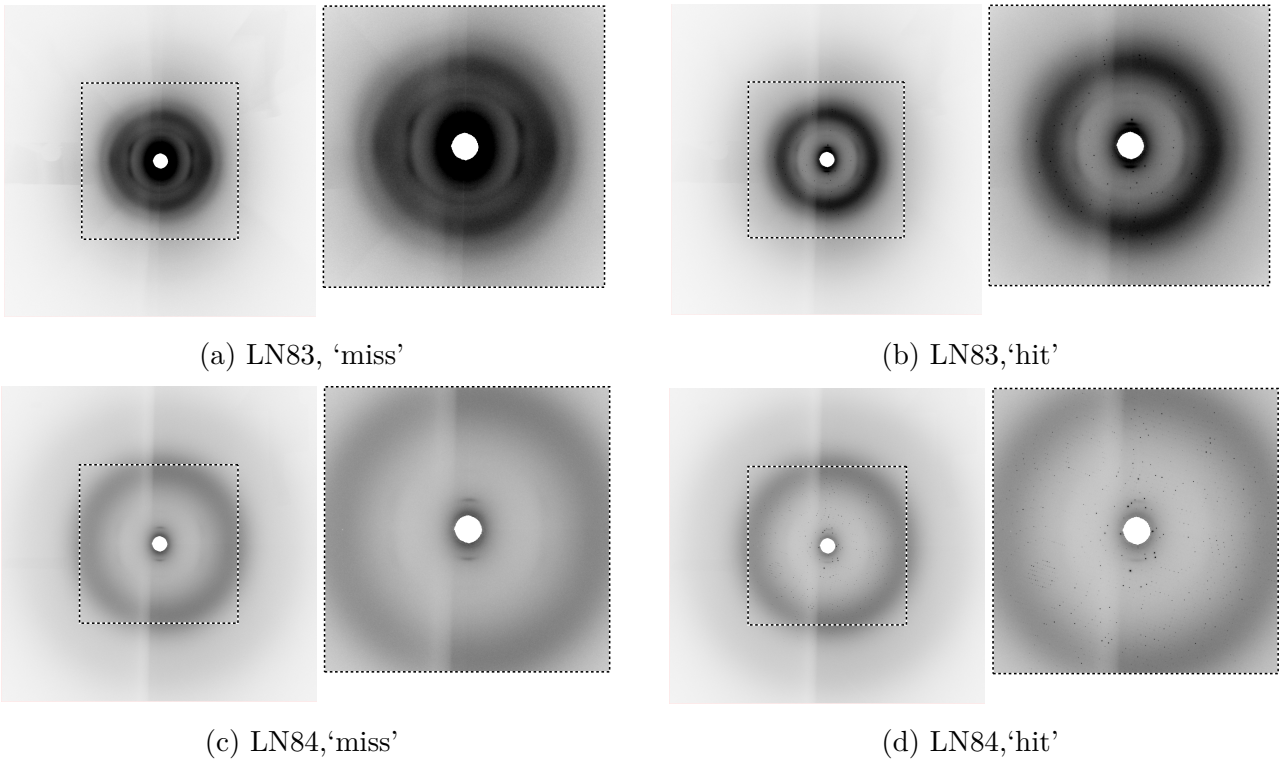


Figure 1: A typical SFX experiment at the European X-ray FEL. The laser produces 10 trains of X-ray pulses per second, with a pulse repetition rate within each train that can vary from 1.1 MHz to 4.5 MHz. The diffraction from the protein sample is measured using an Adaptive Gain Integrating Pixel Detector (AGIPD), which is capable of measuring of up to 352 images from each bunch train at frame rates up to 4.5 MHz [23].



(a) LN83, 'miss'

(b) LN83,'hit'

(c) LN84,'miss'

(d) LN84,'hit'

Figure 2: Representative diffraction patterns randomly selected from the Rayonix detector. In addition, we crop the central region of diffraction patterns to enhance the visibility of Bragg peaks in 'miss' and 'hit' categories. Ke et al. use human annotators to label LN83 and LN84 [9].

Considering these data challenges, sophisticated tools have been developed to process data and provide feedback during experiments [1, 21]. In a typical SFX experiment, the detector may register Bragg peaks from crystal 'hits', otherwise missing in empty shots. Due to the nature of SFX experiments, it is obvious that only samples with Bragg peaks are useful for downstream analysis [23]. Figure 2 shows 'miss' and 'hit' samples randomly selected from a Rayonix detector. Therefore, current methods utilize statistical peak finding to identify and discriminate samples [7, 8]. For example, the *Cheetah* finds and counts the Bragg peaks in each image and keeps the image if this number exceeds some threshold. Finally, the reduced data is output in a facility independent HDF5 output, enabling downstream analysis. For example, *CrystFEL* [21] is employed to view, index, integrate, merge and evaluate diffraction data. Likewise, the *OnDA* [14] provides real time monitoring of data along with experimental conditions.

Over the last decade, deep learning has produced unprecedented breakthroughs in image classification tasks [11]. So, the serial crystallography community has experimented with deep learning methods to classify data and achieve data reduction [2, 9, 20]. Deep learning methods based on Convolutional Neural Networks (CNNs) encode experimental data to classify it into 'hit' or 'miss' categories. While these networks can achieve superior performance on image

classification tasks, their lack of decomposability into individually intuitive blocks makes them hard to understand or interpret [12]. Previous work with CNNs for serial crystallography has inherited these limitations [9]. For example, there is no mechanism to explain how CNNs make decisions while classifying samples into 'hit' or 'miss' categories. Generally, input data in a CNN passes through several layers of multiplication with learned weights and through non-linear transformations in order to make a prediction. Therefore, a prediction may involve millions of mathematical operations depending on the network type. This process makes it challenging for humans to understand the exact mapping from input to prediction; they are a 'black box', see Figure 3. As a result, when such networks fail, they often break down spectacularly without providing meaningful error explanations [12]. Therefore, networks should provide visualizations on their predictions and layers along with standard evaluation metrics.

To this end, the computer vision community has developed methods to visually explain model predictions [13, 18, 19]. Similarly, we present a study using visualization methods from computer vision to highlight data features (attributes) that contribute to the CNN prediction or decision in classifying serial crystallography data into 'hit' or 'miss' categories, using both synthetic and real experimental data. In addition, we use methods to understand what information different layers of a CNN extract from the image. Our qualitative examples reveal that a CNN focuses on discriminative regions of an images while classifying data into 'hit' or 'miss' categories. In other words, it activates different parts for images taken from 'miss' or 'hit' classes.

3 Method

Our goal is to understand with visualization how CNNs classify serial crystallography data into 'hit' and 'miss' categories. Figure 3 shows our pipeline to understand CNN predictions and blocks, which uses both qualitative and quantitative components. In this work, we use two computer vision methods to visualize the function of CNN blocks, and CNN predictions. Generally, visualization methods are applied to supervised neural networks. Therefore, we selected a standard image classification network named AlexNet [10] to provide insights into its behavior. In this section, we provide details on various components of our pipeline including data, neural network and visualization methods and implementation details.

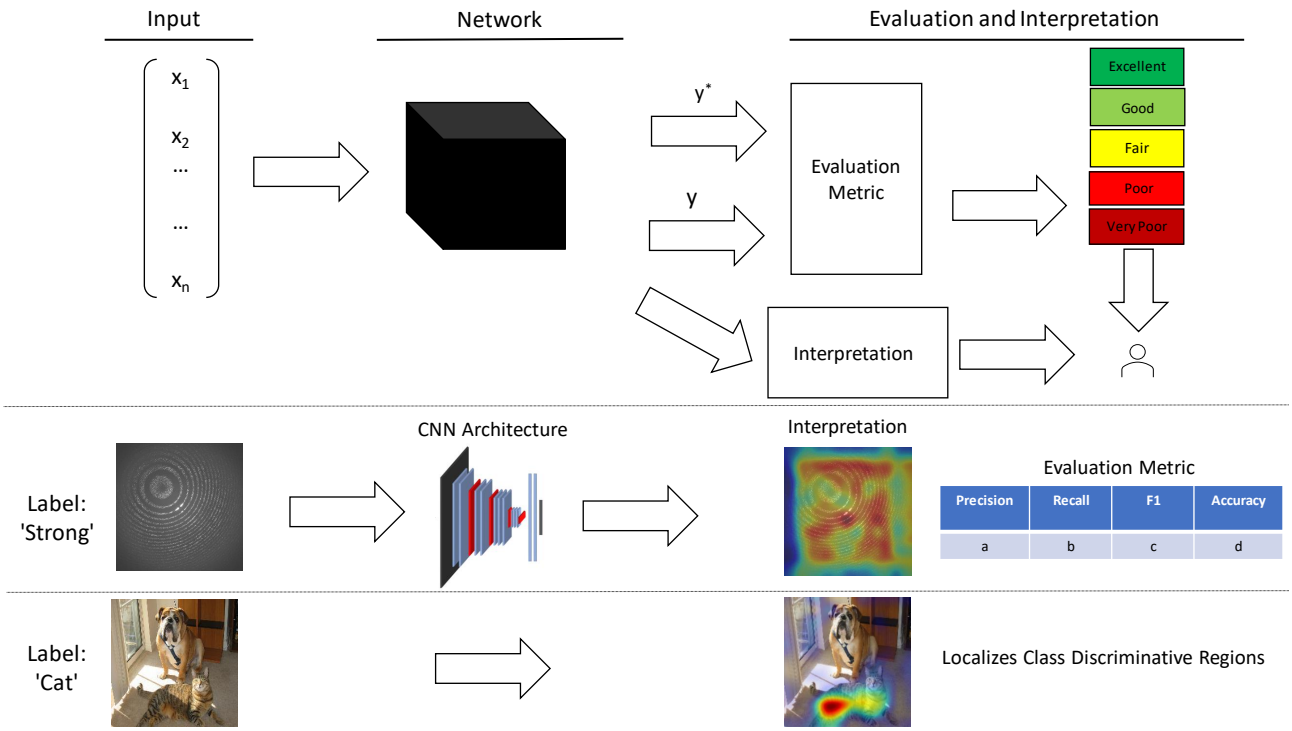


Figure 3: Overview of the proposed methodology to evaluate a deep neural model with interpretation and evaluation metrics. Visual interpretation provides useful information for humans to understand how a deep neural model makes a decision to classify a sample into a certain class while the evaluation metric helps to understand its quantitative performance. For illustration, the figure shows an example with a natural image, where discriminative regions in an image from the 'cat' class are highlighted.

3.1 Datasets

We used synthetic and experimental datasets to visualize CNNs' block representations and predictions. The DiffraNet dataset [20] is comprised of synthetic samples generated using the nanoBragg simulator. The simulator produces different images by taking a single crystal structure and varying the X-ray beam intensity, simulating imperfections in the crystal by breaking it up into smaller crystals, and also varying parameters like the sources of background noise and the orientation of the crystal. DiffraNet consists of $25,000$ samples with an image size of 512×512 divided into five classes: Blank, No Crystal, Weak, Good and Strong. The Blank class denotes images with no X-rays and only detector noise, while in the No-crystal class there is a scattering from amorphous material but no protein crystal. Weak, Good, and Strong represent images with a crystal in the beam with increasingly higher diffracted intensity.

In addition, we used the LN83 and LN84 protein serial crystallography datasets collected at the Macromolecular Femtosecond Crystallography [3] instrument of the Linac Coherent Light Source [22] (LCLS) with the conveyor-belt delivery of crystal specimens [5]. Previous work [9] unpacked the first 2000 images from the the native LCLS data format for further study. Table 1

Table 1: Experimental data.

LCLS dataset (proposal, run)	Incident energy (eV)	Protein	Space group, unit cell (Å)	Instrument	Sample delivery	Detector
LN84, 95	9516	Photosystem II	$P2_12_12_1$, a = 118, b = 223, c = 311	MFX	Conveyor belt	Rayonix
LN83, 18	9498	Hydrogenase	$P2_12_12_1$, a = 73, b = 96, c = 119	MFX	Conveyor belt	Rayonix

shows experimental settings for these two datasets. We used the same images in experiments [9].

3.2 Convolutional Neural Networks

In recent years, deep neural models have remarkably improved state-of-the-art image, video, speech and text processing tasks [11, 17, 16, 15]. One of the fundamental tools leading to these results is a neural network named CNN. Typically, It is composed of several building blocks or layers to transform one volume of activations to another through a differentiable function. We provide brief overview of three layers in a typical CNN i.e. convolution layers, pooling layers and fully connected layers.

A convolutional layer extracts features from the input image or previous layer using the mathematical operation of convolution between the input and a filter of a particular size. The operation is performed with a sliding window over the input image and computing the dot product between the filters and the parts of the input with respect to the size of the filter, producing feature maps. Feature maps provide information on where the features such as edges, corners etc. occur in the image and how well they correspond to the filter. The weights of filters can be trained using gradient descent based algorithms such as stochastic gradient descent. The linear operation of convolution is then followed by applying a nonlinear activation function to each element of the feature map; this makes it possible for the network to learn nonlinear features of the input.

Moreover, a convolutional layer is often followed by a pooling layer which shrinks the size of the convolved feature map to reduce the computational costs. In other words, the pooling operation keeps the detected features in a smaller representation by discarding less significant data at the cost of spatial resolution. The pooling task independently operates on each feature map. The most common methods are max pooling and average pooling, where the former finds the highest value within a window region and discards the remaining values while the latter finds the mean of the values within the region.

A series of convolutional and pooling layers extract increasingly high-level features of the image. These are followed by fully connected layers of neurons which perform classification task. The final fully connected layer has the one output node for each class.

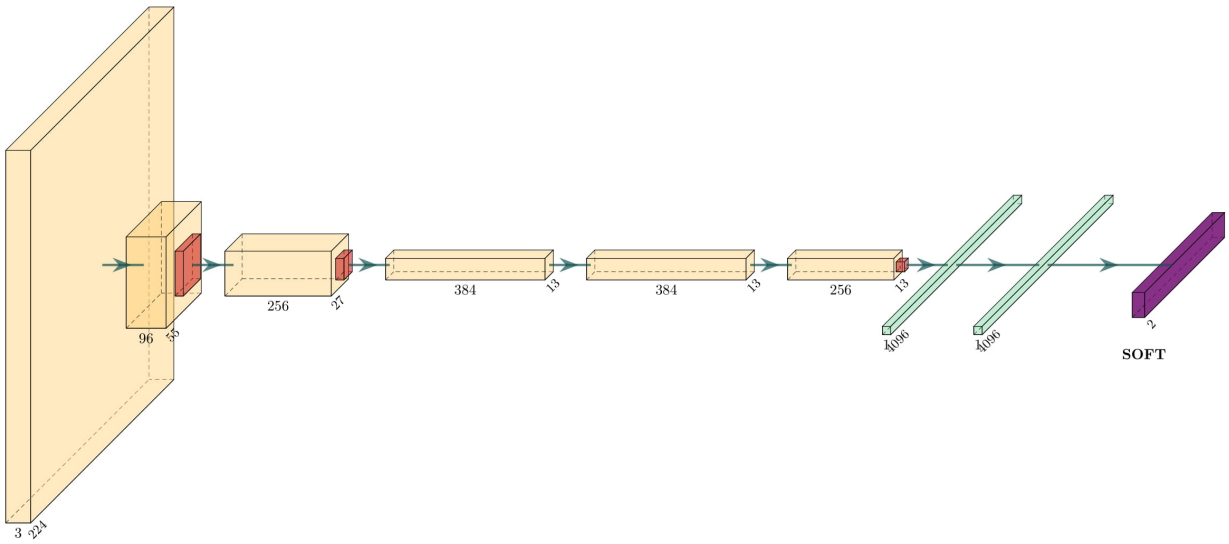


Figure 4: AlexNet architecture. It contains eight layers; the first five are convolutional layers and the last three are fully connected layers. The final fully connected layer has the same number of outputs as the number of classes in the dataset.

In this work, we use a standard benchmark network named AlexNet[10]. It has eight layers, which is fewer than many more recent architectures, but it is still considered a deep neural network. The first five layers are convolutional layers, some of them followed by max-pooling layers, and the last three are fully connected layers, as shown in Figure 4. Moreover, It uses the Rectified Linear Unit activation function which computationally cheap and widely used in CNNs.

3.3 Visualization of Representations

Image representations are a crucial component of almost any image understanding system. It provides information to understand what is encoded by a CNN layer. This is done by taking the output of a CNN layer (referred as representations), and attempting to reconstruct the original input image from it. Later CNN layers contain increasingly high-level information, so we do not expect an accurate or detailed reconstruction, but the reconstruction can indicate what features a layer retains. Fig. 5 shows an overview of the process to reconstruct the original image from the CNN representations. Mathematically, it is formulated as follows:

$$x^* = \text{argmin}_{x \in \mathbb{R}^{H \times W \times C}} l\left(\Phi(x), \Phi_0\right) + \lambda \mathcal{R}(x) \tag{1}$$

where $\Phi(x)$ refers to the representations obtained by passing some image x to the network, and $\Phi(x_0) = \Phi_0$ are the representations obtained from the original image. Moreover, $\mathcal{R} : \mathbb{R}^{H \times W \times C} \rightarrow \mathcal{R}$ is a regulariser capturing an image prior, which is helpful for the reconstruction process by restricting the inversion to the subset of images. Intuitively, it produces an image x^* that is similar to x_o from the representation view point. The process insures that the output is some kind of reasonable image, not just computational noise. The loss function l employed in this work is Euclidean distance as shown below:

$$\ell\left(\Phi(\mathbf{x}), \Phi_0\right) = \left\|\Phi(\mathbf{x}) - \Phi_0\right\|^2 \tag{2}$$

It minimizes the distance between the representations of the original image and reconstructed image. In addition, the regulariser improves the reconstruction process with help of two image prior methods. The first prior used is called α-norm, which is defined as:

$$\mathcal{R}_\alpha(x) = \|x\|_\alpha^\alpha \tag{3}$$

where x is the vectorised and mean-subtracted image. It favours images with a narrower spread of pixel values. The second prior used for discrete image x is called total variation defined as:

$$\mathcal{R}_{V^\beta}(x) = \sum_{i,j} \left((x_{i,j+1} - x_{i,j})^2 + (x_{i+1,j} - x_{i,j})^2 \right)^{\frac{\beta}{2}} \tag{4}$$

where $\beta = 1$. This favours images in which neighbouring pixels have similar values. Finally, the overall objective function is as follows:

$$\left\|\Phi(\sigma\mathbf{x}) - \Phi_0\right\|_2^2 / \left\|\Phi_0\right\|_2^2 + \lambda_\alpha \mathcal{R}_\alpha(\mathbf{x}) + \lambda_{V^\beta} \mathcal{R}_{V^\beta}(\mathbf{x}) \tag{5}$$

where scaling σ is the average Euclidean norm of natural images in a training set and λ_α is the α-norm to encourage the reconstructed image σx to be contained in a natural range. As explained in Section 3.2, typically a CNN detects edges from pixels in first layer(s), then use edges to detect shapes in the next layer(s), and then use it to infer complex shapes and objects in higher or later layers. Thus, the reconstructed image from initial layers may look very similar to the original image.

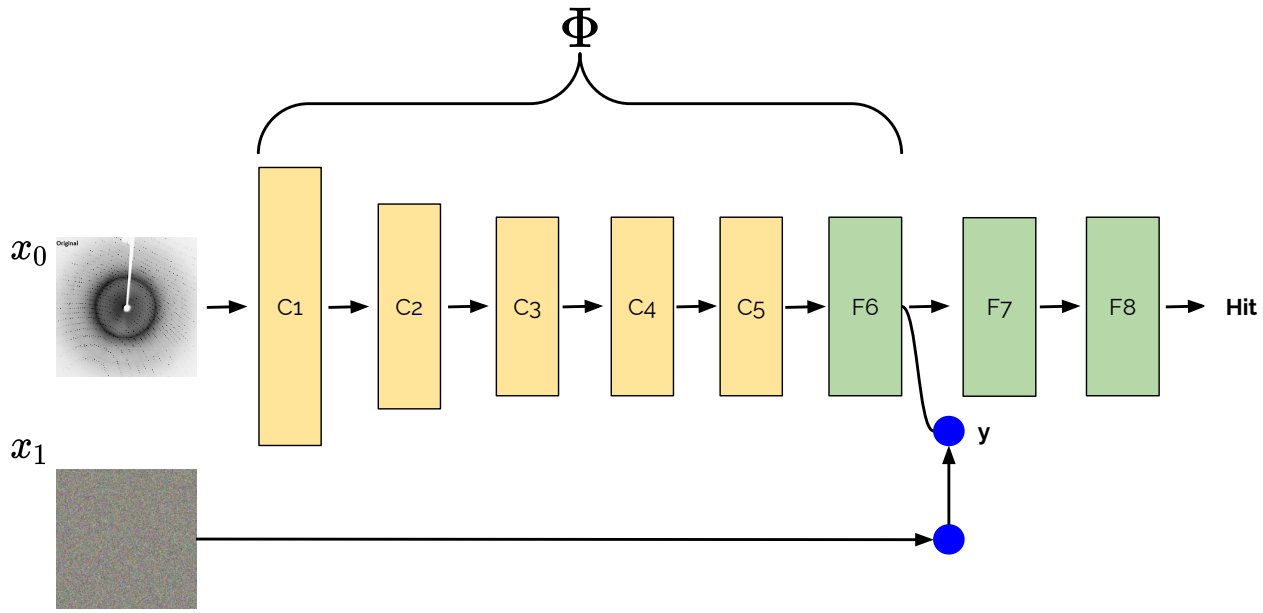


Figure 5: Overview of visualizing CNN block representations [13]. The method starts with a random noise and iteratively reconstructs an equivalent image, demonstrating CNN representation of a block.

3.4 Visual Explanations from Predictions

Visual explanations or interpretations are employed by CNNs to highlight features (attributes) that mostly contribute to a specific prediction. For example, visual explanations can show part(s) or region(s) of the image responsible for a 'dog' prediction with a model trained on dog and cat samples. Likewise, our goal is to visualize part(s) of serial crystallography images responsible for 'hit' or 'miss' classification. To this end, we use Gradient-weighted Class Activation Mapping (Grad-CAM) which requires a differentiable layer generalizing it for a wide variety of CNN architectures [18]. It takes the feature map of the last convolutional layer and multiplies every channel by the gradient of the output class. Afterwards a heat map is generated to highlight activated region(s) of the input image for a specific class. We use the following steps to create heat maps for visual explanations on a specific prediction:

1. Compute the gradient of the score for a specific class y^c (the raw output of the last convolutional layer before softmax) with respect to each of the feature map activations (A^k) of a convolutional layer.

2. Average pool the gradients over the width and height dimension to get the neuron importance weights (α_k^c). This gives us a measure of how strongly each feature map (k)

contributes to an image being classified as a particular class (c).

$$\alpha_k^c = \overbrace{\frac{1}{Z}\sum_i\sum_j}^{\text{global average pooling}} \underbrace{\frac{\partial y^c}{\partial A_{ij}^k}}_{\text{gratients via backprop}} \tag{6}$$

3. Calculate the heatmap by finding the sum of all the feature maps, weighted by their importance and follow it by a ReLU to obtain:

$$L^c_{\text{Grad-CAM}} = \text{Re}\,LU \underbrace{\left(\sum_k \alpha_k^c A^k\right)}_{\text{linear combination}} \tag{7}$$

   ReLU is applied to the linear combination which have positive influence on the specific class, suppressing negative pixels belonging to other class.

4. Reshape and project the maps onto the original image.

Finally, the merged original image and heat map highlight the discriminative region(s) contributing to the classification process.

3.5 Implementation Details

We used the standard set of hyperparameters to train networks in all experiments. Specifically, networks were trained on a graphical processing unit for 50 epochs using a batch-size of 128 with Adam optimizer having exponentially decaying learning rate (initialised to 10^{-5}). In addition, we followed the same train and test splits used in previous work [9, 20]. We evaluated the performance of the network with standard classification evaluation metrics i.e. 'accuracy', 'precision' and 'recall' defined as follows:

$$\text{Precision} = \frac{\text{TP}}{\text{TP+FP}} \tag{8}$$

$$\text{Recall} = \frac{\text{TP}}{\text{TP+FN}} \tag{9}$$

$$\text{Accuracy} = \frac{\text{TP} + \text{TN}}{\text{TP} + \text{TN} + \text{FP} + \text{FN}} \tag{10}$$

where TP is True Positive, TN is True Negative, FP is False Positive, and FN is False Negative.

Table 2: Classification results with AlexNet on DiffraNet dataset.

Method	Accuracy
DeepFreak (GLCM + Random Forest) [20]	98.4
DeepFreak(GLCM + Support Vector Machine) [20]	97.6
DeepFreak [20]	**98.6**
AlexNet (Our implementation)	98.1

Table 3: DiffraNet confusion matrix for the test set with AlexNet.

		Blank	No Crystal	Weak	Good	Strong	Recall(%)
True Class	Blank	2069	0	0	0	0	100
	No Crystal	2	3266	0	0	0	99.9
	Weak	3	24	3273	46	0	97.8
	Good	0	0	62	2341	41	95.8
	Strong	0	0	0	60	1412	95.9
	Precision(%)	99.9	99.3	98.1	95.7	97.1	

4 Experiments

In this section, we provide an overview of datasets and implementation details along with qualitative and quantitative results. In our experiments, we visualize CNN representations along with region(s) important for making a specific prediction. The aim of these experiments is to understand the internal working of a standard CNN (AlexNet) while classifying serial crystallography data.

In the first experiment, we trained a standard AlexNet on the simulated DiffraNet dataset, (Table 2). Our implementation produced competitive performance compared to the DeepFreak. Moreover, we used a confusion matrix to summarize the prediction results, (Table 3). We observed that misclassification often occurs between 'Weak' and 'Good' and 'Good' and 'Strong' categories. These results indicate that neighbour classes are similar. Our implementation achieves perfect quantitative performance on 'Blank' and 'No Crystal' categories. For the purposes of data reduction, which is the main aim, we do not care about weak vs good vs strong, as long as we get hit vs non-hit correct. Thus, we can discard diffraction patterns from these categories, achieving meaningful data reduction. The quantitative performance (accuracy) indicates that the model is successfully classifying the synthetic images. Similarly,

Table 4: Classification results (accuracy) on real experimental datasets with AlexNet.

Method	Datasets	
	LN83	LN84
Ke et al. [9]	**96.0**	**90.0**
AlexNet (Our implementation)	82.2	87.0

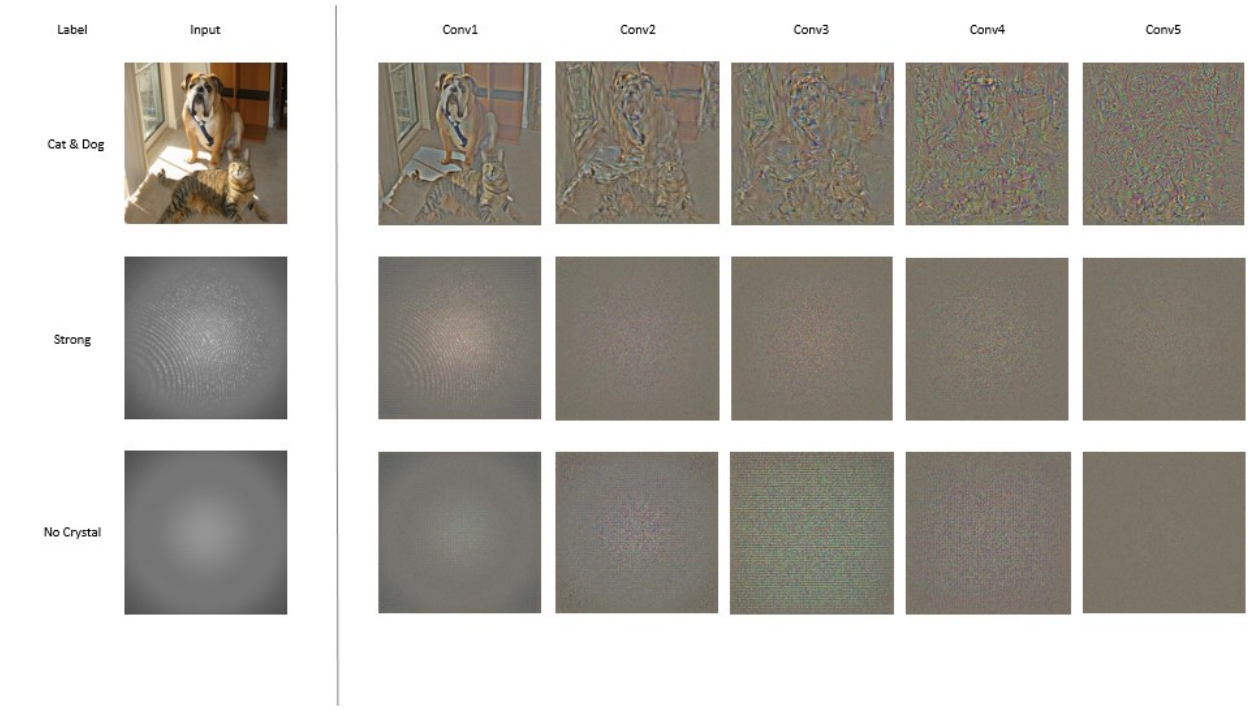


Figure 6: Three examples of CNN representations at the first five convolutional layers of AlexNet. The network detects edges from pixels in first layer, then use edges to detect shapes in the next layer, and then uses it to infer complex shapes and objects in later layers. Thus, despite growing fuzziness, convolutional layers continue to maintain photographically accurate representations of input images. (Best viewed in color and zoomed in)

we perform the classification experiments with real experimental data consisting of two diverse datasets, (Table 4). Our implementation produced slightly lower accuracy compared to the previous work (Ke et al [9]). Evaluations numbers do not provide insights on how a CNN classifies images from serial crystallography. Thus, we use qualitative methods to provide insights with the following two experiments:

- Analyze CNN representations of various layers with the serial crystallography data
- Visualize discriminative regions while classifying the serial crystallography data into 'hit' or 'miss' categories

We extracted representations from various layers of the model and used them to reconstruct the original image, as described previously. Figure 6 shows the reconstruction of images from different classes ('Strong' and 'No Crystal') along with a natural image. We added a natural image example in order to understand qualitative visualizations. The first few layers are significantly similar to the input images. Qualitative results indicate that a CNN detects edges from pixels in the first layer. In addition, all convolutional layers maintain a photographically faithful representation of the input image, although with increasing fuzziness. In addition, re-

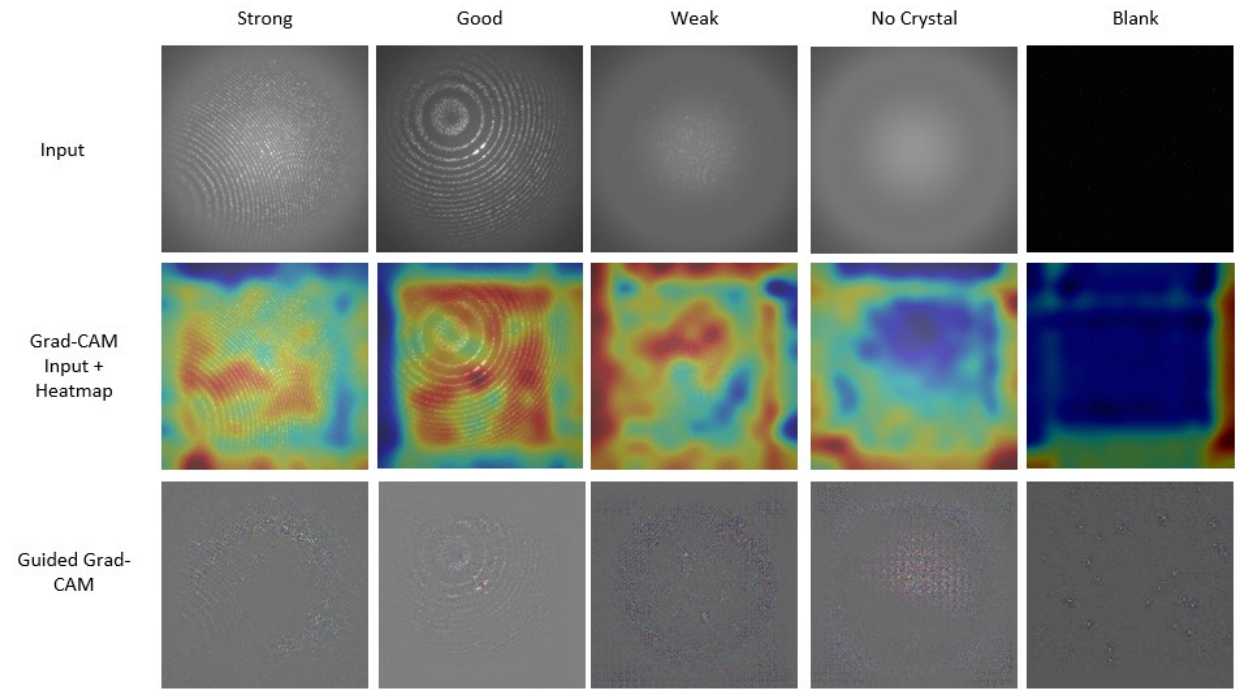


Figure 7: Grad-CAM visualization of five classes of DiffraNet, highlighting contributing features. Guided Grad-CAM are added to highlight fine-grained details. (Best viewed in color and zoomed in)

constructed images of 'Strong' and 'No Crystal' diffraction patterns are distinctively dissimilar indicating that the deep neural model extract 'unique' representations for each of them.

Moreover, we visualize important parts by localizing the discriminative parts of samples, (Figure 7). Heat maps generated with Grad-CAM show that the deep neural model localizes distinct regions while classifying input images into 'Strong', 'Good', 'Weak', 'No Crystal' and 'Blank' categories. Moreover, we added Guided Grad-CAM visualization to provide fine-grained details like pixel-space gradient visualization. We observed that activations or focused areas are increased from 'Blank' to 'Strong' classes. Furthermore, we merged 'Blank' and 'No Crystal' classes into 'miss' and 'Weak', 'Good' and 'Strong' classes into 'hit' with an aim to visualize discriminative regions, see Figure 8. These visualizations show that a deep neural model encodes distinct regions for various classes. Ke et al. [9] argue that a CNN exploits Bragg peaks heuristics for classification task. However, our qualitative results indicate that the network encodes both Bragg peaks and background of the input samples. Intuitively, a CNN focuses on regions of the image where Bragg peaks are present, if there are any. As explained earlier that the DiffraNet dataset contains synthetic data with high quality images which made discriminative regions more prominent in the visualization.

Finally, we visualize discriminative regions for experimental datasets including LN83, LN84, LO19, L498. We selected random samples from the 'hit' and 'miss' categories, see Figure 9.

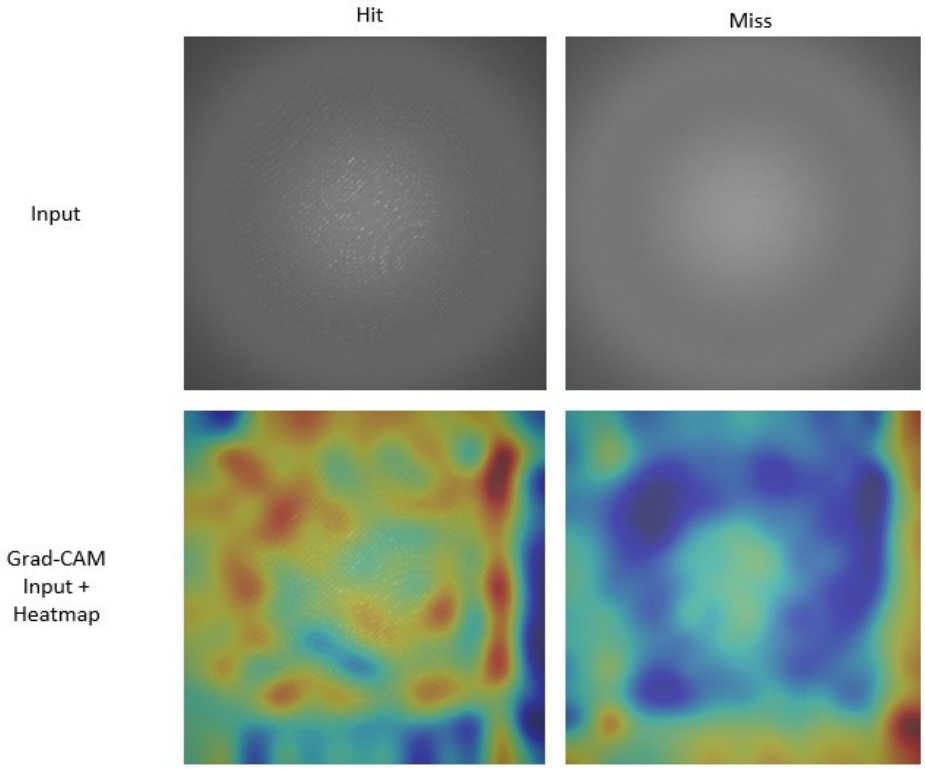


Figure 8: Grad-CAM and Guided Grad-CAM visualization of experimental datasets, highlighting contributing features. (Best viewed in color and zoomed in).

We observe that the 'hit' image contains higher activations compared to a 'miss' for two experimental datasets (LN83 and LN84).

5 Conclusion

In recent years, massive amounts of experimental data have been produced in serial femtosecond crystallography at X-ray free-electron laser facilities. Although these datasets are large, only a fraction of the data is useful for later analysis. Thus there has been interest in using Convolutional Neural Networks to process serial crystallography data. Convolutional Neural Networks successfully categorize this data into the intended categories (hit and miss), but previous research has not explained how these networks achieve this results, making them a 'black box'. Thus, we have presented a qualitative and quantitative study to visualize representations and discriminative regions significant to classify serial crystallography data. Our study reveals that a CNN encodes both Bragg peaks and background to classify an image into 'hit' or 'miss' category.

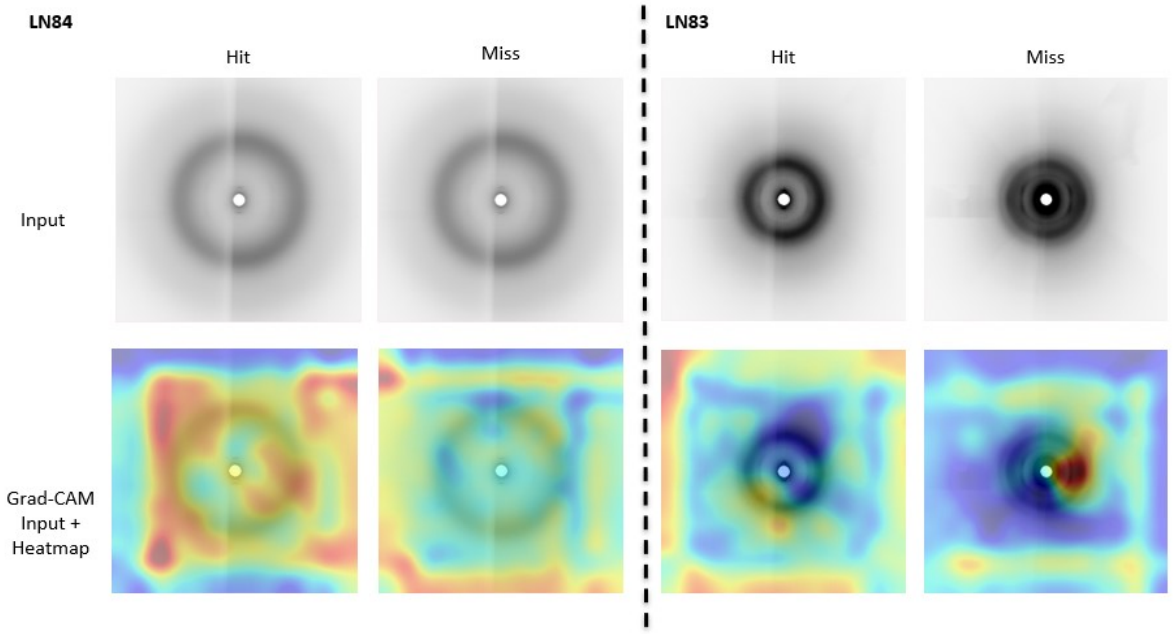


Figure 9: Grad-CAM and Guided Grad-CAM visualizations with experimental datasets (LN83 and LN84), highlighting contributing features. (Best viewed in color and zoomed in).

6 Acknowledgements

We acknowledge 'Helmholtz IVF project InternLabs-0011 (HIREX)' and 'Helmholtz Innovationspool project Data-X' for providing funds to carry out the research. In addition, the work was supported in part through the Maxwell computational resources operated at Deutsches Elektronen-Synchrotron DESY, Hamburg, Germany Moreoever, we appreciate Thomas White detailed feedback on the work.

References

[1] Anton Barty, Richard A Kirian, Filipe RNC Maia, Max Hantke, Chun Hong Yoon, Thomas A White, and Henry Chapman. Cheetah: software for high-throughput reduction and analysis of serial femtosecond x-ray diffraction data. *Journal of applied crystallography*, 47(3):1118–1131, 2014.

[2] Daniel Becker and Achim Streit. A neural network based pre-selection of big data in photon science. In *2014 IEEE Fourth International Conference on Big Data and Cloud Computing*, pages 71–76. IEEE, 2014.

[3] Sébastien Boutet, Aina E Cohen, and Soichi Wakatsuki. The new macromolecular femtosecond crystallography (mfx) instrument at lcls. *Synchrotron radiation news*, 29(1):23–28, 2016.

[4] Henry N Chapman, Petra Fromme, Anton Barty, Thomas A White, Richard A Kirian,

Andrew Aquila, Mark S Hunter, Joachim Schulz, Daniel P DePonte, Uwe Weierstall, et al. Femtosecond x-ray protein nanocrystallography. *Nature*, 470(7332):73–77, 2011.

[5] Franklin D Fuller, Sheraz Gul, Ruchira Chatterjee, E Sethe Burgie, Iris D Young, Hugo Lebrette, Vivek Srinivas, Aaron S Brewster, Tara Michels-Clark, Jonathan A Clinger, et al. Drop-on-demand sample delivery for studying biocatalysts in action at x-ray free-electron lasers. *Nature methods*, 14(4):443–449, 2017.

[6] John N Galayda. The lcls-ii: A high power upgrade to the lcls. Technical report, SLAC National Accelerator Lab., Menlo Park, CA (United States), 2018.

[7] Marjan Hadian-Jazi, Marc Messerschmidt, Connie Darmanin, Klaus Giewekemeyer, Adrian P Mancuso, and Brian Abbey. A peak-finding algorithm based on robust statistical analysis in serial crystallography. *Journal of Applied Crystallography*, 50(6):1705–1715, 2017.

[8] Marjan Hadian-Jazi, Alireza Sadri, Anton Barty, Oleksandr Yefanov, Marina Galchenkova, Dominik Oberthuer, Dana Komadina, Wolfgang Brehm, Henry Kirkwood, Grant Mills, et al. Data reduction for serial crystallography using a robust peak finder. *Journal of Applied Crystallography*, 54(5), 2021.

[9] T-W Ke, Aaron S Brewster, Stella X Yu, Daniela Ushizima, Chao Yang, and Nicholas K Sauter. A convolutional neural network-based screening tool for x-ray serial crystallography. *Journal of synchrotron radiation*, 25(3):655–670, 2018.

[10] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. *Advances in neural information processing systems*, 25:1097–1105, 2012.

[11] Yann LeCun, Yoshua Bengio, and Geoffrey Hinton. Deep learning. *nature*, 521(7553):436–444, 2015.

[12] Zachary C Lipton. The mythos of model interpretability: In machine learning, the concept of interpretability is both important and slippery. *Queue*, 16(3):31–57, 2018.

[13] Aravindh Mahendran and Andrea Vedaldi. Understanding deep image representations by inverting them. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 5188–5196, 2015.

[14] Valerio Mariani, Andrew Morgan, Chun Hong Yoon, Thomas J Lane, Thomas A White, Christopher O'Grady, Manuela Kuhn, Steve Aplin, Jason Koglin, Anton Barty, et al. Onda: online data analysis and feedback for serial x-ray imaging. *Journal of applied crystallography*, 49(3):1073–1080, 2016.

[15] Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781*, 2013.

[16] A. Nagrani, J. S. Chung, and A. Zisserman. Voxceleb: a large-scale speaker identification dataset. In *INTERSPEECH*, 2017.

[17] Muhammad Saad Saeed, Muhammad Haris Khan, Shah Nawaz, Muhammad Haroon Yousaf, and Alessio Del Bue. Fusion and orthogonal projection for improved face-voice association. In *ICASSP 2022-2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 7057–7061. IEEE, 2022.

[18] Ramprasaath R Selvaraju, Michael Cogswell, Abhishek Das, Ramakrishna Vedantam, Devi Parikh, and Dhruv Batra. Grad-cam: Visual explanations from deep networks via gradient-based localization. In *Proceedings of the IEEE international conference on computer vision*, pages 618–626, 2017.

[19] Karen Simonyan, Andrea Vedaldi, and Andrew Zisserman. Deep inside convolutional networks: Visualising image classification models and saliency maps. In *In Workshop at International Conference on Learning Representations*. Citeseer, 2014.

[20] Artur Souza, Leonardo B Oliveira, Sabine Hollatz, Matt Feldman, Kunle Olukotun, James M Holton, Aina E Cohen, and Luigi Nardi. Deepfreak: learning crystallography diffraction patterns with automated machine learning. *arXiv preprint arXiv:1904.11834*, 2019.

[21] Thomas A White, Richard A Kirian, Andrew V Martin, Andrew Aquila, Karol Nass, Anton Barty, and Henry N Chapman. Crystfel: a software suite for snapshot serial crystallography. *Journal of applied crystallography*, 45(2):335–341, 2012.

[22] William E White, Aymeric Robert, and Mike Dunne. The linac coherent light source. *Journal of synchrotron radiation*, 22(3):472–476, 2015.

[23] Max O Wiedorn, Dominik Oberthür, Richard Bean, Robin Schubert, Nadine Werner, Brian Abbey, Martin Aepfelbacher, Luigi Adriano, Aschkan Allahgholi, Nasser Al-Qudami, et al. Megahertz serial crystallography. *Nature communications*, 9(1):1–11, 2018.