Peak finding for Crystallography

Po-Nan Li
Department of Electrical Engineering
Stanford University
liponan@stanford.edu

Chun Hong Yoon [*]
Linac Coherent Light Source
SLAC National Accelerator Laboratory
yoon82@slac.stanford.edu

Abstract

We present a Bragg peak detection system based on YOLO architecture for X-ray crystallography. The 8-layer convolutional neural network we developed outperforms the original Yolo2 in Bragg peak detection and achieves similar performance of Yolo3 but with much fast speed and simpler architecture.

1. Introduction

Crystallography is a powerful technique that enables the determination of the inner structure of nanomolecules such as proteins. In a typical crystallographic setting, an X-ray with very short wavelength (e.g. 0.1 nm) is shined on the crystal, whose diffraction pattern, effectively its Fourier transform intensity, is collected by an area detector [1]. Due to the repeated units in the crystal, the Fourier transform has several significantly noticeable bright pixels, namely Bragg peaks. Figure 1 illustrates the work flow in a typical crystallographic experiment: the detector records the Bragg peaks, which are located and used to retrieve the crystal's parameters. In order to retrieve the structural information, the crystallographic data processing pipeline [2] needs to first locate all of these Bragg peaks. Unlike other object recognition problem, the detection of Bragg peaks is complicated by that 1) Bragg peaks are fine features that are only up to 7×7 pixels large; 2) precise coordinates of Bragg peaks are required, i.e. prediction error < 1 pixel; 3) There is only one class of object, i.e. the peak; 4) the intensities of pixels span across several orders of magnitude, making the feature normalization very difficult. Such task is currently done by an exhaustive search algorithm that goes through every pixel, column by column and row to row, to find the locations of peak pixels. Although the state-of-the-art peak finder can precisely locate the peaks on a diffraction image in less than a second if the user provide reasonable parameter, these parameters are, however, sometimes difficult to finetune.

LCLS (Linac Coherent Light Source at SLAC National Accelerator Laboratory), one of the largest free-electron laser facilities in the world, generates TB of data everyday [1], but little machine learning approaches have been applied to utilize or explore such "big data." The pioneering work by Ke *et al.* demonstrates the application of deep learning to the X-ray crystallography data, but the usage is limited to classifying the data as hit, miss or in between [3]. The work by Park *et al.* also employs deep learning for classification problem, but for crystal structures [4]. Their work outperforms the human-level, but the application to the serial femtosecond crystallography (SFX) experiment is unexplored.

Here, we study the application of deep learning approach to the SFX data by developing a convolutional neural network (CNN) [5] that can detect all of the Bragg peaks and return the coordinates thereof in one shot. Our network is based on the architecture of YOLO2 (you only look once), a unified CNN-based object detection systems [6, 7, 8], but with significant modification to fit the uniqueness of peak detection problem, i.e. pixel-level fine features. We use the diffraction images from the LCLS experiment CXIC0415, which is public dataset for research and academic purposes.

2. Data

We used a subset of diffraction image dataset from a LCLS SFX experiment on complex of Streptavidin and Selenobiotin [9] and split the data into three independent sets for this project [2]. Table 1 details the numbers of diffraction images and Bragg peak instances in the training, validation and test datasets. Each diffraction image is composed by 64 separate pixel array modules, each has 194×185 pixels, which are carefully arranged to receive maximum amount of diffraction. Data of these module are store in a compact format, a 1552×1480 16-bit gray-scale image. Each

---

[*]non CS231n-student author

[1]For example, the full volume of CXIC0415, the dataset we use for this project, totals 18 TB.

[2]The full dataset is publicly available via the Coherent X-ray Imaging Data Bank (CXIDB) at http://www.cxidb.org/id-54.html, although we instead accessed the data via an internal interface [2].
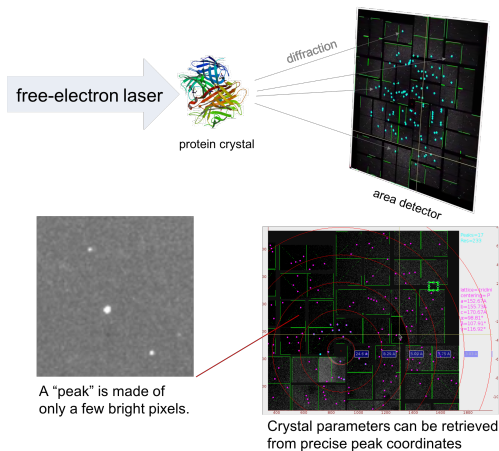
Figure 1. (Top) Crystallographers reconstruct the structure of proteins by collecting the diffraction image, which has a number of Bragg peaks, and locating all of the peaks. (Bottom) Bragg peaks are two-dimensional features that are made of few pixels.

Table 1. Number of diffraction images and Bragg peak instances in each dataset. All datasets have the same distribution. 1% of data are kept for validation (0.5%) and testing (0.5%).

Dataset	Train	Val	Test	Total
# images	199,675	1,000	1,000	201,675
# peaks	12,498,810	63,543	63,917	12,626,270

image is accompanied by a list of peak coordinates which were found by a brute-force algorithm. The raw pixel values in each image indicate the ADUs (analog-to-digital units), which is approximately proportional to the number of photons a pixel detects. We clamp the pixel value at $10,000$ and re-quantize to 8-bit and output all diffraction image images to PNG files, as we found this preprocessing largely reduces the data overhead. The codes for preprocessing the LCLS experimental data are developed for this project and are available on the GitHub repository[3].

3. Method

We designed a new YOLO architecture which is dedicated to detect the crystallographic Bragg peaks by combing the strengths of three generations of YOLO net, i.e. YOLOv1 [6], YOLOv2 [7] and YOLOv3 [8]. One of the well-known weakness of the YOLO family is that YOLO often fails to detect small features as YOLO uses a number of maxpooling to reduce the number of parameters. Although the authors of YOLO already address this issue in YOLOv3 by using a number of shortcuts, i.e. ResNet architecture [10] and pyramid detection [11], we still think the excessive number of striding is harmful for detecting small features like Bragg peaks. Figure 2 shows our proposed

---

[3]https://github.com/leeneil/peaknet

16-layer CNN architecture for YOLO detection, which has only two maxpooling layers, in the hope to preserve as many pixel-level features as possible. The first maxpooling layer reduces the image size and the second maxpooling layer aims to provide multi-scale features. The output features before and after the second maxpooling layers are then concatenated by using a shortcut and a "reorg" layer, which rearranges 2×2 pixels into a 1×1 pixel with 4 feature channels. The final 30 channels are responsible for the YOLO detection, which needs the x, y coordinates and width and height of each prediction box, and its objectness (the probability that there is an object) and the probability of the class. In this case, we only have one class, which is the Bragg peak. Five sets of prior are used to facilitated the predictions, therefore $5 \times (4 + 1 + 1) = 30$ features are needed for each pixel at the output layer. Later we will present alternative design in the hope to increase processing speed but without sacrificing the performance. The implementation uses Darknet as the deep learning framework [12] and custom Python codes for experimentation [4].

For training, we set a fixed box size of 11×11 so all the labeled Bragg peak instances have the same size. Setting a finite box size also ensures that there are sufficient features to be learned by the CNN. We compute the IOU (intersection over union) for each predicted box with its nearest ground truth neighbor. The predicted box is considered as successful detection if the IOU is larger than 0.5. We then calculate the sensitivity, defined by the number of true positives (TP) divided by the number of true positives and false negatives (FN) combined, and the precision, defined by the number of TP divided by the number of TP and FP combined. Although IOU is useful for determining if a detection is successful, in order to quantify the precision of location prediction, we also calculate the average location error for each true positive.

4. Experiments

The ultimate goal of the peak detection is twofold: the precision of detection and the accuracy of the peak location must meet the standard of crystallography requirement; on the other hand, the processing speed should be also high so the detection system might be run on-line, i.e. enabling real-time analysis when an experiment is being conducted. To this end, we will focus on the detection precision while ensuring the number of operation is just sufficient to avoid unnecessary burden of computations. Table 2 shows the performance of all network architectures considered. All tests were performed on a standard Google Cloud Compute instance with a nVidia K80 graphic processing unit (GPU). Architecture of PeakNet is same as the one illustrated in

---

[4]We used Darknet instead of other popular frameworks simply because it is written in C and runs significantly faster than others
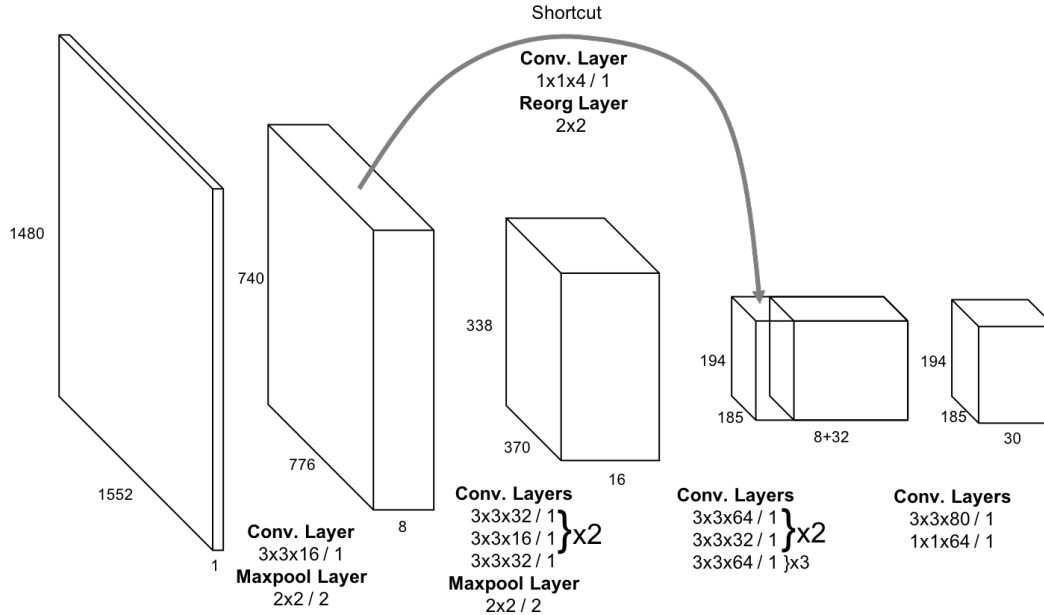
Figure 2. Our proposed CNN architecture. A shortcut provides feature maps with less downsampling. The output volumes has size of 194×185 and 30 features for YOLO detection. The number following the slash indicates the stride number.

Figure 2. PeakNet-Shallow has fewer alternating convolutional layers but same number of filters. PeakNet-Small has the same depth as PeakNet but only one fourth of filter number. Lastly, PeakNet-Fat has same depth as PeakNet-Shallow but four times number of filters as PeakNet. All architecture were trained on the training set for 6,240 iterations (roughly 1 epoch) with mini-batch size of 32 and learning rate of 0.001. Figure 3 shows the precision vs. sensitivity plot on the validation data. It is noticeable that PeakNet-Shallow works generally worse than PeakNet, while PeakNet-Small shows somehow decent precision at the cost of low sensitivity, presumably due to the fact the nature of less filters and hence less over-fitting. Finally, PeakNet-Fat achieves nearly Yolo3 level performance, but has less distance error and higher speed. Therefor PeakNet-Fat is chosen for final testing. Figure 4 shows the detection result on a representative image from the testing set. The trade-off between precision and sensitivity can be easily leveraged by turning the detection threshold.

5. Conclusion

We designed and trained a Bragg peak detection system which is based on YOLO2. The presented network outperforms YOLO2 in detecting Bragg peaks and achieve same performance as YOLO3, which is much deeper and complex. We experimented several alternative network designs and compare their performance by using the metrics of sensitivity, precision and speed. The future works include op-

Table 2. Comparison of all architectures studied. Detection threshold is fixed to 0.25 for all architectures. Spe(ed): average time needed to process a image in milliseconds. Sen(sitivity): TP/(TP+FN). Pre(cision): TP/(TP+FP). Dis(tance): average distance between the true and predicted peak location. IOU: average IOU. Dis. and IOU are conditioned on successful detection, i.e. IOU > 0.

Architecture	Spe	Sen	Pre	Dis	IOU
PeakNet	465	0.076	0.880	0.82	0.15
PeakNet-Shallow	557	0.004	0.787	1.08	0.18
PeakNet-Small	347	0.134	0.945	0.75	0.23
PeakNet-Fat	760	0.683	0.831	0.82	0.30
Yolo3	845	0.820	0.740	0.85	0.35

timizing the network design, e.g. number of convolutional layers and number of filters, for maximum precision/speed ratio, and to train the network on different experimental data to further generalize the detection system.

Author contributions

Chun Hong Yoon of Linac Coherent Light Source, SLAC National Accelerator Laboratory conceived the idea of using yolo to detect Bragg peaks. Po-Nan Li processed the data, conducted the research and wrote the report with inputs from C.H. Yoon.
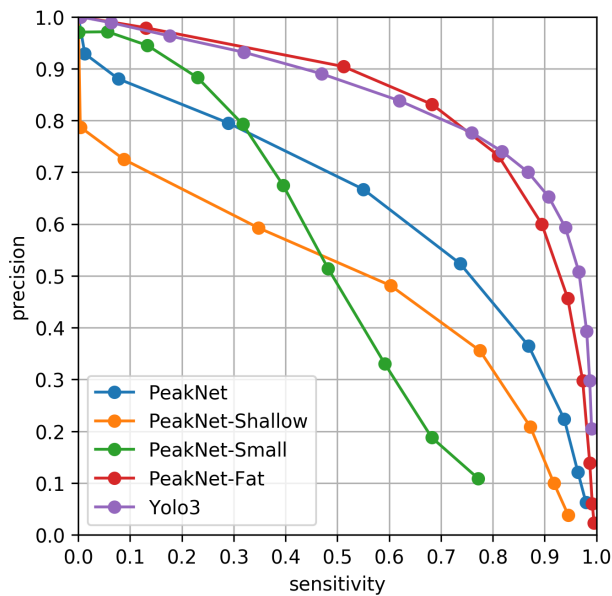
Figure 3. Precision vs. sensitivity comparison of all architectures considered. Precision/sensitivity is tuned by changing the detection threshold, from 0.005 to 0.5.

Acknowledgements

References

[1] J. Tenboer *et al.*, Science **324**, 1246 (2014).

[2] D. Damiani, M. Dubrovin, I. Gaponenko, W. Kroeger, T.J. Lane, A. Mitra, C.P. O'Grady, A. Salnikov, A. Sanchez-Gonzalez, D. Schneider, and C.H. Yoon, "Linac Coherent Light Source data analysis using psana," J. Appl. Crystallogr. **49**, 672 (2016).

[3] Tsung-Wei Ke, Aaron S. Brewster, Stella X. Yu, Daniela Ushizima, Chao Yang and Nicholas K. Sauterb, "A convolutional neural network-based screening tool for X-ray serial crystallography," J. Synchrotron Rad. 25, 655 (2018).

[4] Woon Bae Park, Jiyong Chung, Jaeyoung Jung, Keemin Sohn, Satendra Pal Singh, Myoungho Pyo, Namsoo Shin, and Kee-Sun Sohna, "Classification of crystal structure using a convolutional neural network," IUCrJ 4, 486 (2017).

[5] Y. LeCun, L. Bottou, Y. Bengio and P. Haffner, Proc. IEEE **86**, 2278 (1998).

[6] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, "You only look once: Unified, real-time object detection," arXiv:1506.02640 (2015).

[7] J. Redmon and A. Farhadi, " YOLO9000: Better, Faster, Stronger," arXiv:1612.08242 (2016).

[8] J. Redmon and A. FarhadiarXiv, "YOLOv3: An Incremental Improvement." arXiv:1804.02767 (2018).

[9] M. S. Hunter *et al.*, "Selenium single-wavelength anomalous diffraction de novo phasing using an X-ray-free electron laser," Nat. Comm. 7, 13388 (2016).

[10] Shaoqing Ren, Kaiming He, Ross Girshick, Jian Sun, " Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks," arXiv:1506.01497 (2015).

[11] Tsung-Yi Lin, Piotr Dollr, Ross Girshick, Kaiming He, Bharath Hariharan, Serge Belongie, "Feature Pyramid Networks for Object Detection," arXiv:1612.03144 (2016).

[12] J. Redmon, "Darknet: Open source neural networks in C," http://pjreddie.com/darknet/ (2013 - 2016).
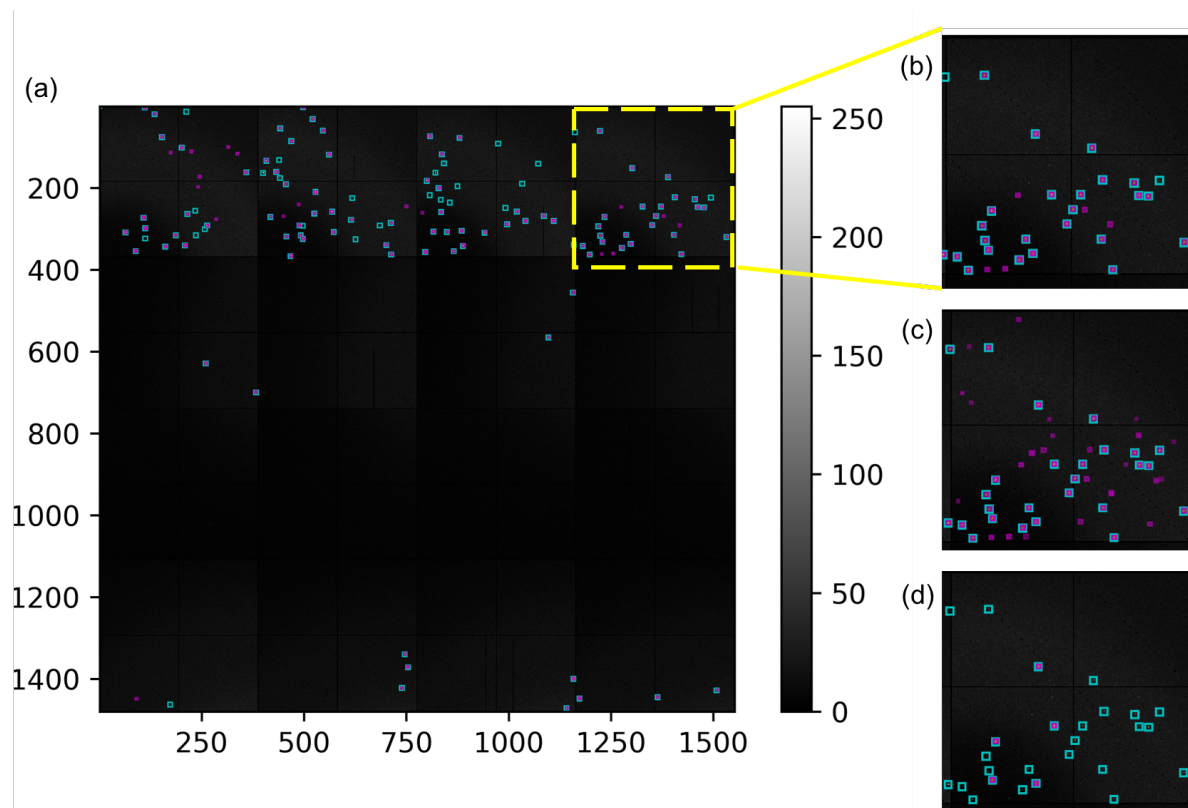
Figure 4. Peak detection on a diffraction image from the testing set. (a) Cyan and pink boxes mark the Bragg peaks of ground truth labels and predicted by the CNN, respectively. In this case, the CNN achieves sensitivity of 0.79, precision of 0.85 and error distance of 0.84 at threshold of 0.25. (b) Zoom in of the yellow box region in (a). (c) The region as (b), but threshold value is 0.1. The excessive number of false positives drops the precision to 0.46. (d) The region as (b), but threshold value is 0.4. The precision is 1 as every detection is a true positive, but at the cost of very low sensitivity, in this case 0.24.