Proceedings of the Thirtieth AAAI Conference on Artificial Intelligence (AAAI-16)

Learning Deep Convolutional Neural Networks for X-Ray Protein Crystallization Image Analysis

Margot Lisa-Jing Yann and **Yichuan Tang**
Department of Computer Science
University of Toronto
{lyan, tang}@cs.toronto.edu

Abstract

Obtaining a protein's 3D structure is crucial to the understanding of its functions and interactions with other proteins. It is critical to accelerate the protein crystallization process with improved accuracy for understanding cancer and designing drugs. Systematic high-throughput approaches in protein crystallization have been widely applied, generating a large number of protein crystallization-trial images. Therefore, an efficient and effective automatic analysis for these images is a top priority. In this paper, we present a novel system, CrystalNet, for automatically labeling outcomes of protein crystallization-trial images. CrystalNet is a deep convolutional neural network that automatically extracts features from X-ray protein crystallization images for classification.

We show that (1) CrystalNet can provide real-time labels for crystallization images effectively, requiring approximately 2 seconds to provide labels for all 1536 images of crystallization microassay on each plate; (2) compared with the state-of-the-art classification systems in crystallization image analysis, our technique demonstrates an improvement of 8% in accuracy, and achieve 90.8% accuracy in classification. As a part of the high-throughput pipeline which generates millions of images a year, CrystalNet can lead to a substantial reduction of labor-intensive screening.

Introduction

Crystallization is a bottleneck problem in structural biology. A protein's structure determines its function, and similarly structured proteins have similar functions. Obtaining a protein's structure is crucial to the understanding of its functions and interactions with other proteins. Protein X-ray crystallography is a methodology used to discover protein structure. However, structural genomics is a complex, expensive and failure-prone process, where chemical conditions for each protein's successful crystallization are difficult to find. Typically, crystallization proceeds as a trial-and-error approach. Systematic high-throughput (HTP) approaches lead to rapid collection of trial information regarding the crystallization problem. X-ray diffraction can only take place after proteins have crystallized. During HTP protein crystallization trials, each well of an assay plate constitutes a unique crystallization experiment, where each of them is filled with the target protein solution and a unique cocktail of crystallizing reagents. An image is captured periodically over each well, to record an individual protein's different stages, changing from clear solution to crystal or precipitate. This process produces a large number of images which require image processing and stage classification.

An automatic image scoring system is needed to substantially replace/assist human experts to identify successful crystallized images during HTP screening process. Machine learning and computer vision techniques have been widely applied to solve the crystal detection problem including image analysis, feature extraction and classification. However, automatic crystal detection in practice is a very challenging problem. The challenges are the large amounts of images, low resolution of images, variations of lighting and other conditions, and the diversity of protein crystal morphology.

In this paper, we use an image dataset (Snell et al. 2008b; 2008a) from the Hauptman-Woodward Institute, whose Hydra robots set up microbatch-under-oil crystallization experiments in 1536-well microassay plates (Luft, Snell, and DeTitta 2011). Thousands of proteins screened each year with this 1536-well system, and it produces millions of images recorded which require humans to manually examine the crystallization stages. Thus, one critical step for scaling up the HTP pipeline is the automatic labeling of protein crystallization-trial images, for example, into classes such as crystal, clear or precipitate. This paper focuses on learning a convolutional deep neural network to efficiently and accurately analyze protein crystallization images automatically in HTP pipelines.

Related Background and Seminal Works

Many research groups have utilized various machine learning techniques to tackle this needle-in-a-haystack problem, as only a few crystals appear among thousands of trials. An automatic system is required to correctly reject clear or precipitate images, yet accurately identify the crystal image for X-ray diffraction. Two aspects need to be considered which relate to the factors determining the image classification performance: (1) each group analyzes different types of images, pixel quality, color versus gray scale; (2) the different number of class categories. The better the image quality, the relatively easier it is for classification; the more detailed the class categories, the more difficult it is to achieve accurate classification. (While all groups used different datasets, they are all grayscale, imaged from the assay plates.)

The methodologies in previous works can be divided into two major groups. One group focuses on the extraction of distinctive and useful features from crystallization images while using standard off-the-shelf classification algorithms. The other group uses standard image processing techniques to obtain simpler features but focus on learning and combining classifiers.

Bern et al. (2004) combine edge detection with dynamic programming to track drop boundaries and further extracted promising features for five-class classification: Empty, Clear, Precipitate, Microcrystal Hit, and Crystal. This five-class classifier achieved about 12% false negative and about 14% false positives on the test data (total 2113 human-annotated images were obtained, and each image is 700*750 8-bit gray-scale pixels, with a 2*2 mm field of view.) Liu, Freund, and Spraggon (2008) built a system using open-source image analysis components to extract hundreds of marginally discriminative features and applied a boosting algorithm to learn alternating decision trees to identify 'Harvestable' (crystal size $> 10\mu m$) and 'Crystal Hit'. Tested on coarse-screen and fine-screen experiments, the coarse system achieved a mean ROC-AUC score of 0.93. Cumbaa and Jurisica (2010) extracted extensively 12375 features from images and developed a 10-way classifier and 3-way classifier using Random Forests to score collected images, where 20% false-negative rate for crystals was reported in the clear/has-crystal/other 3-way classifier and 80% of crystal-bearing images were successfully recognized. For membrane protein crystallization processes, UV images are typically obtained. Automated scoring images to identify large, small or scattered crystals still have not reached the accuracy of conventional human inspections (Kissick et al. 2010; Haupert and Simpson 2011). Hung et al. (2014) developed a two-tier cascade classifier using naive Bayes and Random Forest classifier, while using Elastic Nets to select representative heterogeneous image features extracted using Gabor filters and Fourier transformation, obtaining 74% accuracy over 3-categories. In Sigdel, Pusey, and Aygun (2013), multilayer perceptrons are applied to assist crystallographers in scoring trial images efficiently (less than 3 seconds to classify one image) and achieved 1.2% crystal error rate with an overall accuracy of 88%.

Automated labeling systems enable researchers to run more experiments in shorter time and reduce the human labor to manually speculate about each image for each trial. However, it is challenging to develop a system to have a low rate of false negatives (missed crystals), but not at the cost of increasing the number of false positives. A support vector machine (SVM) (Pan et al. 2006) was developed and achieved less than 4% false negative rate but at the cost of 40% false positive rate. Elsey and Wirth (2014) used an approach of involving the image segmentation segmentation of crystal images. However, crystal detection in images remains a very challenging problem due to the varying sizes, shapes and types.

In this paper, we introduce a novel application of deep convolutional neural networks - CrystalNet - to automatically extract task-specific features and robustly handle variations, leading to accurate and efficient classification of protein crystallization images.

Methods

In this section, we provide the architecture of CrystalNet, the parameter learning procedure and the details of the algorithm. Starting with a grayscale image of a protein crystallization trial, a convolutional neural network is trained to compute intermediate features, then followed by the final estimation of class probabilities for each image.

CrystalNet Architecture

In machine learning, artificial neural networks (NNs) are models capable of nonlinear regression and classification, distinguishing among K different classes. Specifically, Convolutional Neural Networks (CNNs) (LeCun et al. 1998) are NNs where weight-sharing occurs at certain layers of the network. The weight-sharing allows the model to achieve equivariance representation of the underlying pixel data. In addition, pooling layers provide invariance to image transformations by reducing spatial resolution via downsampling. Recently, CNNs have been successfully applied to large-scale image classification task (Krizhevsky, Sutskever, and Hinton 2012) achieving state-of-the-art in their respective fields. We illustrate a CNN in Figure 1.

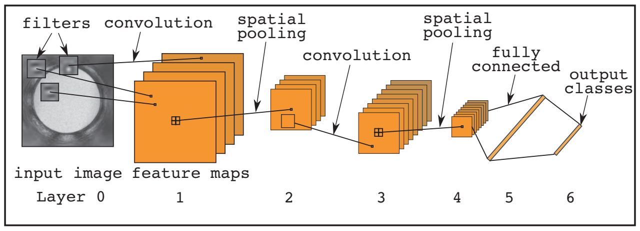

Figure 1: *Diagram of the standard convolutional neural network.*

In a CNN, filters or weights act like pattern detectors that are applied across the input image. A single map of the layer 1 feature maps are the activation responses from a single filter. 2D location information is preserved by the convolution operation. Spatial pooling provides a measure of invariance to input variations via downsampling. Deep CNNs refers to the depth of the computation graph. While Figure 1 has only 6 layers, more layers can be added. In particular, the convolution (C) layer followed by spatial pooling (S) layer can be combined into a two-layer module. Additional 'C-S' modules can be stacked before layer 4 in Figure 1. Other image preprocessing and normalization layers are also introduced in our proposed CrystalNet, with 13 hidden layers in total. Table 1 contains the detailed description of the 13-layer architecture of CrystalNet.

Convolution layer Let a 3D array of feature maps or inputs be X with size $w_x \times h_x \times m_x$, where w_x is the width, h_x is the height and m_x is the number of feature maps. For a gray image input, $m = 1$. Let the output Y with size $w_y \times h_y \times m_y$. Let $\mathbf{x}_j$ be a 2D slice of X with size $w_x \times h_x$,

Layer	Type	Latent Variables	Filter Size	# Weights
0	Image	size:128×128 maps:1	-	-
1	CN	size:128×128 maps:1	-	-
2	Horiz	size:128×128 maps:1	-	-
3	Transf	size:112×112 maps:1	-	-
4	Conv	size:112×112 maps:64	5×5	1600
5	Pooling	size:56×56 maps:64	2×2 max	-
6	Conv	size:56×56 maps:64	5×5	102K
7	Pooling	size:28×28 maps:64	2×2 max	-
8	Conv	size:28×28 maps:128	5×5	204K
9	Pooling	size:14×14 maps:128	2×2 max	-
10	Conv	size:6×6 maps:128	3×3	147K
11	FC	2048	-	9.4M
12	FC	2048	-	4.2M
13	Output	10	-	20K

Table 1: *Model architecture of the proposed CrystalNet: Layer 0 is the input image (# of maps corresponds to the # of color channels); CN refers to a local contrast normalization layer; Horiz: horizontal mirroring layer; Transf: 2D similarity transformation layer; Conv: convolution layer; Pooling: spatial max-pooling layer; FC: fully connected layer; Output layer size is 10, depending on the # of classes in the classification task.*

$\mathbf{y}_k$ be a 2D slice of Y with size $w_y \times h_y$, and ω_{jk} be a 2D filter of size $d \times d$. The convolution layer computes the following:

$$\mathbf{y}_k = g\left(\sum_j \omega_{jk} * \mathbf{x}_j\right) \tag{1}$$

where $*$ performs 2D filtering across the input $\mathbf{x}_j$ and $g(\cdot)$ is a nonlinearity such as a sigmoid, hyperbolic tangent, or rectified linear (Eq. 2).

Pooling layer There are two common types of pooling operations, max and average. With max-pooling, 2×2 neighboring nodes of a hidden feature map compute the max among themselves and the max value is sent to the higher layer. With average-pooling, the average of the 2×2 neighborhood is computed instead of the max. The higher layer will have 1/4 as many nodes. The pooling operation can provide invariance to noise and jitter since small shifts within the pooling neighbors would not affect higher layer activations. Invariance to intra-class variations allows models to generalize and perform better on the test set.

Contrast Normalization layer This layer performs subtractive and divisive normalization locally. Within every 7×7 patch, the mean of the patch and the variance are computed. Each patch then subtracts away its mean and is divided by its variance. This normalization process is very useful for dealing with illumination changes in the image.

Horizontal Mirroring layer Due to the fact that mirrored crystal images do not change the class it belongs to, we can potentially double the training set by adding mirrored images of all of the training images. Adding more images can surely make the classifier more robust, at the cost of space redundancies. Instead of preprocessing the dataset by adding all mirrored images, we randomly (with probability of 0.5) apply horizontal reflection to input images during training. This allows training to use all images and 50% of their mirrored image without storing twice the number of images.

Transformation layer Following the same idea as mirroring, the class label stays the same regardless of what translations, rotation and scale changes the input image undergoes. Therefore, in this layer, we apply small 2D Similarity transformations (Szeliski 2011) to the image in order to train effectively on a much larger training set. Enlarging the dataset in this way is a common "trick" in computer vision and deep learning literature and provides a substantial boost in generalization performance.

Output layer The output layer typically has softmax nodes in 1-of-K codes. For 10-way classification, the output layer contains a 10 dimensional vector of all zeros except a single '1'. The position of the '1' in the vector indicates the class. The standard cross-entropy training objective (Bishop 1994) is used with the softmax output layers.

Neuron nonlinearity After the convolution operation for the convolution layers or matrix multiplication for the fully connected layers, an element-wise nonlinearity is applied, which is necessary for a nonlinear classifier. While well-known nonlinearities such as the sigmoid and hyperbolic tangent functions are possible, we use the recently discovered rectified linear units (Nair and Hinton 2010), which have been found to facilitate faster training. Let x be the activation before nonlinearity and y be the activation after rectified linear activation, we have

$$y(x) = \begin{cases} x & \text{if } x > 0 \\ 0 & \text{if } x \leq 0 \end{cases} \tag{2}$$

Procedure

The algorithm for classification in CrystalNet starts with the preprocessing of all input images by taking the central square crop and resizing to a resolution of 128×128. Since the images are grayscale, the number of maps or channels is 1 (Layer 0 in Figure 1). Contrast normalization followed by horizontal reflection are performed. Next in Layer 3, image transformation by randomly shifting the image ± 8 pixels, rotating ± 60 degrees, and applying a scaling factor of 0.8 to 1.2. These transformations essentially augments the dataset with more examples and lead to better model generalization. Scaling factor of 1.2 performs "zoom-out" and make the original image appear smaller. To remove the extra border regions caused zooming out, subsequent layers only process the 112×112 central pixels (see Table 1). The first convolution (Layer 4) has 64 feature maps and 64 different filters. The activation function is computed according to Eq. 1 and this nonlinear operation of Eq. 2 is applied after the convolution. Max-pooling is applied in Layer 5 to reduce the resolution by 2. Alternating convolution and pooling operations are performed until Layer 11, when all hidden nodes have connection to 2048 next layer nodes, computed using matrix multiplication followed by rectified linear function of Eq. 2. Finally a softmax is taken at the final output layer giving a probability vector which sums to 1.0.

Learning

During learning, we seek to optimize the parameters (also known as weights) of CrystalNet to perform better under the cross-entropy objective. Cross-entropy loss is commonly used as a surrogate to the classification error because it is continuous and closely approximates the classification error. We use the first-order optimization method known as Stochastic Gradient Descent (SGD) (Robbins and Monro 1951) for learning. In practice, SGD performs extremely well with sublinear convergence and is widely used in practice (Bottou 2010).

The actual learning process divides the entire training set into small minibatches of 128 images each. Without using minibatches, each parameter update would take place only after averaging the gradient of all images, resulting in slow convergence per gradient computation. At the other extreme, updating the parameters based on the gradient of only 1 image is very noisy. Empirically, minibatch sizes from 100 to 500 have shown to work well for a wide range of datasets. Let $\{\mathbf{x}_b, \mathbf{y}_b\}$ be the training data, $\mathbf{x}_b$ is the input data and $\mathbf{y}_b$ represent true labels from minibatch b. The partial derivatives of the objective w.r.t. to the current parameters are first computed. Let $L_b(f(\mathbf{x}_b), \mathbf{y}_b)$ be the cross-entropy loss, where $f(\mathbf{x}_b) \in (0, 1)$ is prediction probability value. α is the learning rate, and μ is the momentum which smooths out the stochasticity caused from using minibatches and helps the training converge faster. We update the momentum by: $\mu = 0.9\mu - \alpha \nabla_\theta L_b(f(\mathbf{x}_b), \mathbf{y}_b)$. Then, the weight parameters θ are then updated using the new momentum: $\theta_{t+1} \leftarrow \theta_t + \mu$.

Besides the contrast normalization layer (see Table 1: Layer 1 CN), a per pixel normalization is also performed in two steps: first, the mean value across the entire training set is subtracted; second, each pixel is then divided by its standard deviation across the training set. Normalization helps first-order optimization converge faster by reducing the curvatures of the objective function. As learning progresses, the validation accuracy will gradually plateau and the parameters will converge to a local minimum.

While the number of adaptive parameters of our model is in the millions whereas the number of training images is only in the tens of thousands, the unique convolutional architecture of CrystalNet ameliorates the overfitting problem. CrystalNet demonstrates highly accurate classification of protein crystallization images.

Experimental Results and Analysis

To demonstrate the effectiveness of using deep convolutional neural networks for the automated classification protein crystallization, we trained CrystalNet on the dataset (Snell et al. 2008b; 2008a) from the Hauptman-Woodward Institute. This dataset contains 163,894 high-resolution grayscale labeled images of protein crystallization trials on 1536-well plates. This data is manually labeled and consists of two subsets. One subset has 147,456 images generated from testing 96 proteins, each with 1536 cocktail recipe trials. The other subset has 16,438 images all containing crystals. The labels are obtained through the meticulous work of human experts, where each image has been labeled by one or more experts for different states of individual protein crystallization. Images are labeled with seven possible attributes, which describe the outcome of the experiment: *clear*, *phase separation*, *precipitate*, *skin effect*, *crystal*, *junk/garbage*, and *unsure*. The *clear* drop condition denotes the absence of the other six attributes. In Figure 2, we show sample images from each of the seven attributes.

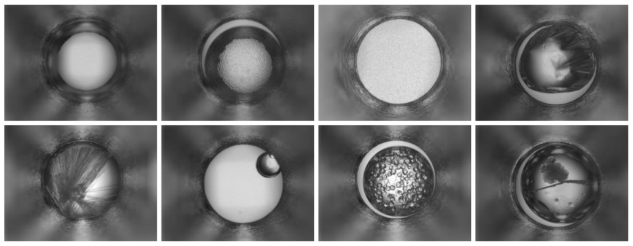
Figure 2: *Examples of protein crystallization image databases. From top to bottom, left to right: clear, phase separation, precipitate, skin effect, crystal, junk/garbage, unsure, and unsure.*

Human experts can disagree on the difficult borderline images, leading to some label noise in the dataset. Of the 163,894 images, 85,188 have unanimous labels. This subset of 85,188 "clean" images is used in our experiments. We randomly divide 80% of the clean images into a training set and 20% into a validation/test set.

We preprocess each grayscale image by cropping the central square and resizing it to a resolution of 128×128 pixels. Standard data normalization per pixel is applied as described in Section Learning. Our model CrystalNet uses the architecture described in Table 1. During training we use a learning rate α of 0.01 and perform 150,000 parameter updates in total. Momentum μ is set at 0.9, and the L_2 regularization constant β is set at 0.0001. The above learning hyperparameters are found by first performing several short trial runs with a subset of the training data, and selecting the ones which obtained the lowest error. The weight parameters are all randomly initialized by sampling from a Gaussian distribution with zero mean and 0.01 standard deviation. (See Section Computation Time for training time required.) After training, Figure 3 shows examples of the first layer filters. These filters extract interesting features useful for protein crystallography classification.

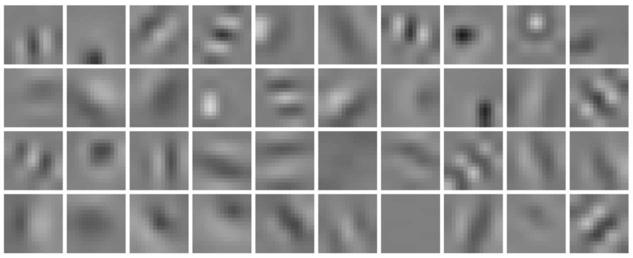
Figure 3: *Examples of the first layer filters learned by our deep convolutional neural net. These filters have resemblances to engineered feature extractors such as edge, spatial-frequency, texture and interest point detectors from computer vision.*

By setting the output layer size of CrystalNet to be the desired number of classes (e.g. K), a K-way classifier can be trained. We trained a 10-way classifier in our experiments, $K = 10$. In the following section, we report a significant performance boost by CrystalNet compared to previous work for 10-way classification.

10-way Classification Results

Due to the nature of the protein crystallization process, multiple physical reactions can co-occur in a crystallization process, such as precipitate, crystal, skin and phase. With certain attributes appearing in combination, 10 class types are defined by crystallographers to capture the fine stage changes during the protein crystallization process, as follows: 1. Clear, 2. Precipitate, 3. Crystal, 4. Phase, 5. Precipitate and Crystal, 6. Precipitate and Skin, 7. Phase and Crystal, 8. Phase and Precipitate, 9. Skin, 10. Junk. Table 2 presents the confusion matrix of the 10-way classifier performance on the validation set.

Model Variations

Deep convolutional neural networks have a plethora of hyperparameters governing the model architecture and data preprocessing. These hyperparameters affect both the computation time and the accuracy of the classifier. We investigate several main variations of the hyperparameters in this section, and demonstrate the performance on the same validation dataset on CrystalNet 10-way classifier. Insights gained can be used to balance the trade-off between speed (both learning and testing phase) and performance.

Training set size The total number of training data used is relatively large at 68,155. However, the number of images required to learn a good performance classifier is a function of both the complexity of the dataset as well its redundancy. We empirically explore the performance of our classifier trained on subsets of total training data. By holding all other learning hyperparameters the same, we train classifiers with varying number of training images and evaluate the performance on the same validation set. In the experiment, the evaluating training set number starting from approximately 10% of the total training set, 6817, increases 10% (6817) each time. As shown in Figure 4, the performance on the validation set increases as the number of training images increases, and it reaches approximately saturation when about 40,000 images are used for training.

Image resolution Higher resolution images maintain finer details which are useful for discrimination. However, higher resolution also requires higher computational resources and can make optimization more difficult. We empirically explore the validation accuracy of the 10-way classifier as a function of the size of the images used. Lower resolution data are generated using linear interpolation and anti-aliasing. Table 3 shows accuracy rates and average time of three runs on the same validation set but with different resolution images, from 256×256 to 32×32.

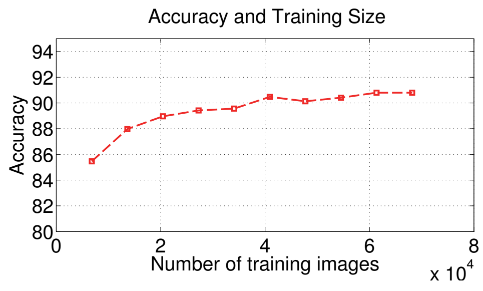

Figure 4: *Performance on the same validation set as a function of the number of unique training images. While gradually increasing, performance stays roughly constant after 40,000 training images.*

Classification Analysis

The key challenging aspect of an automatic labeling image system is to have a low rate of false negative (missed crystals), but also keep the number of false positives low. Table 4 presents CrystalNet's crystal-detection performance in the test set. We generate these ROC plots by using the 10-way classifier where 1 class is crystal and the other classes are aggregated into the non-crystal class. Figure 5 shows that CrystalNet produces an area under curve (AUC) 0.9903 for crystal class classification. By achieving a high recall and high precision rates, which can provide a very low rate of missed crystal images, and the majority of images (precipitate) would not need to be examined manually. This is a significant improvement compared with (Pan et al. 2006), where a similar true positive resulted in 40% false positives.

Comparison We compared CrystalNet with the standard image classification pipeline approach. The standard approach first requires the manual design of features such as edge detectors, spatial/radial frequency filters and texture-based features. Classifiers such as Random Forests (RF) can subsequently be trained on top of these engineered features. In contrast, CrystalNet *learns* to use its many layers for extracting useful features specifically for this given task. Using the same dataset, we evaluate and compare to the previous state-of-the-art approach of (Cumbaa and Jurisica 2010), where 10% of the extracted 12,375 rich image features are used to train a Random Forest classifier and a deep neural net classifier. We also evaluate and compare to anther baseline method, Nearest Neighbor, with the same images as inputs (note that the training and validation split are the same for all models.) Table 5 shows the 10-way classification overall accuracy and the time consumption for predictions on the validation set. CrystalNet achieves an absolute 8% improvement in overall accuracy compared to the RF approach. While Nearest Neighbor is computationally fast, it is almost 20% less accurate than CrystalNet.

The class distribution in the 10-way classification is imbalanced. A different way to evaluate performance is to plot crystal vs. non-crystal classification rate. Figure 5 shows that the ROCs of crystal class classification are produced using CrystalNet, deep Neural Net and RF.

[1]The Random Forest and Neural Net classifier execution time does not include time taken to extract image features.

Labels	Clear	Precipitate	Crystal	Phase	Precipitate & Crystal	Precipitate & Skin	Phase & Crystal	Phase & Precipitate	Skin	Junk	Total	Recall
Clear	5831	9	2	26	0	0	0	0	2	7	5877	0.9922
Precipitate	29	5563	6	38	34	54	0	6	0	2	5732	0.9705
Crystal	65	20	1069	19	167	3	41	0	5	2	1391	0.7685
Phase	51	31	10	1007	11	1	6	2	1	1	1121	0.8983
Precipitate & Crystal	2	258	135	12	864	37	31	0	0	0	1339	0.6453
Precipitate & Skin	0	85	0	1	4	776	0	0	7	0	873	0.8889
Phase & Crystal	3	6	90	67	65	4	157	1	0	0	393	0.3995
Phase & Precipitate	0	29	0	13	7	3	1	30	0	0	83	0.3614
Skin	12	0	0	0	0	6	0	0	116	2	136	0.8529
Junk	30	4	1	0	0	0	0	0	0	53	88	0.6023
Total	6023	6005	1313	1183	1152	884	236	39	131	67	17033	-
Precision	0.9681	0.9264	0.8142	0.8512	0.7500	0.8778	0.6653	0.7692	0.8855	0.7910	Overall:	0.9080
AUC	0.9990	0.9937	0.9863	0.9938	0.9725	0.9957	0.9751	0.9754	0.9969	0.9975	-	-

Table 2: *Confusion matrix for the 10-way classifier on validation dataset: each class category's classification accuracy is given. Each row i represent the number count for the true category type i, element ij is the number of images misclassified as category j, and element ii is the total count number of images correctly classified as true class type i. (80% for training, 68,155 images; where 20% for validating, 17,033 images.) For each category classification's ROC, its area under curve (AUC) is given in the last row.*

Resolution	256×256	128×128	64×64	32×32
Accuracy(%)	90.5	90.8	86.5	78.5
Avg. time(s)	528	300	114	32

Table 3: *Performance on a function of image resolution.*

Labels	no-crystal	has-crystal	Total	Recall
no-crystal	13830	80	13910	0.9942
has-crystal	506	2617	3123	0.8380
Total	14336	2697	17033	-
Precision	0.9647	0.9703	Overall:	0.9656

Table 4: *Confusion matrix for crystal bi-classifier: 17033 images*

10-way	CrystalNet	Neural Net	RF[1]	Nearest Neighbor
Accuracy(%)	90.8	86.9	82.8	70.7
Time (secs.)	300	1	209	286

Table 5: *Performance comparison of CrystalNet, Neural Net, RF and Nearest Neighbor.*

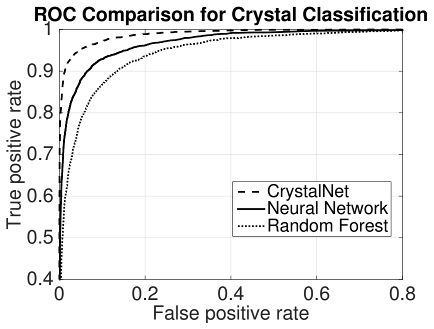

Figure 5: *ROC curve for bi-classification of crystal*

Computation Time

The implementation of CrystalNet is in Compute Uniform Device Architecture (CUDA) GPU code by nVidia. High level Matlab wrappers are also used for CrystalNet training. The training takes approximately 1.5 days on a single GeForce GTX Titan board. The GPU board includes 6 GB of on board memory, 2688 CUDA cores and a GPU clock of 876 MHz. Training phase includes 150,000 weight updates which amounts to around 300 passes over the entire training data. We also benchmarked the time required for CrystalNet to classify images after training. The time spent to classify one 128×128 image is approximately 86 milliseconds, while the time spent to classify 128 images is approximately 181 milliseconds. This increased computation time is nonlinear due to the CUDA implementation which can exploit the parallelism existing within each batch of 128 images.

Discussions and Future Work

We have presented and evaluated a deep convolutional neural network, CrystalNet, for the labeling of protein crystallization images during different phases. CrystalNet learns to extract task-specific features from protein crystallization images directly, without resorting to the manual designing of features. We demonstrate that accurate protein crystallization pipelines are not impossible: CrystalNet can provide labels for images generated during the HTP process effectively, with a low miss rate and high precision for crystal detection. Moreover, CrystalNet operates in real-time, where labeling all 1536 images on each single plate only requires approximately 2 seconds. The combination of accuracy and efficiency makes a fully automated HTP crystallography pipeline possible, substantially reducing labor intensive screening. Future work involves to apply CrystalNet to a large dataset of proteins for laboratory usage. To complete the automation of the pipeline, it is also necessary to develop a recommendation system of chemical reagents for each protein crystallization process. Furthermore, we plan to investigate applying CrystalNet to other types of images, such as ultraviolet and microscope images, exploring the practical use of CrystalNet in other application domains.

Acknowledgments

We are grateful for the support, assistance and comments provided by our lab members and the anonymous reviewers for helpful suggestions.

References

Bern, M.; Goldberg, D.; Stevens, R. C.; and Kuhn, P. 2004. Automatic classification of protein crystallization images using a curve-tracking algorithm. *Journal of Applied Crystallography* 37(2):279–287.

Bishop, C. M. 1994. Mixture density networks. Technical Report NCRG/94/004, Aston University.

Bottou, L. 2010. Large-Scale machine learning with stochastic gradient descent. In *Proceedings of COMPSTAT 2010*, 177–186. Physica-Verlag HD.

Cumbaa, C. A., and Jurisica, I. 2010. Protein crystallization analysis on the World Community Grid. *Journal of Structural and Functional Genomics* 11(1):61–69.

Elsey, M., and Wirth, B. 2014. Fast automated detection of crystal distortion and crystal defects in polycrystal images. *Multiscale Modeling & Simulation* 12(1):1–24.

Haupert, L. M., and Simpson, G. J. 2011. Screening of protein crystallization trials by second order nonlinear optical imaging of chiral crystals (SONICC). *Methods* 55(4):379–386.

Hung, J.; Collins, J.; Weldetsion, M.; Newland, O.; Chiang, E.; Guerrero, S.; and Okada, K. 2014. Protein crystallization image classification with elastic net. In *Proceedings SPIE 9034, Medical Imaging 2014: Image Processing*.

Kissick, D.; Gualtieri, E.; Simpson, G.; and Cherezov, V. 2010. Nonlinear optical imaging of integral membrane protein crystals in lipidic mesophases. *Analytical Chemistry* 82(2):491–497.

Krizhevsky, A.; Sutskever, I.; and Hinton, G. 2012. Imagenet classification with deep convolutional neural networks. In *Advances in Neural Information Processing Systems 25*. The MIT Press. 1106–1114.

LeCun, Y.; Bottou, L.; Bengio, Y.; and Haffner, P. 1998. Gradient-based learning applied to document recognition. *Proceedings of the IEEE* 86(11):2278–2324.

Liu, R.; Freund, Y.; and Spraggon, G. 2008. Image-based crystal detection: a machine-learning approach. *Acta Crystallogr., Sect D: Biol. Crystallogr.* 64(12):1187–1195.

Luft, J. R.; Snell, E. H.; and DeTitta, G. T. 2011. Lessons from high-throughput protein crystallization screening: 10 years of practical experience. *Expert Opinion on Drug Discovery* 6(5):465–480.

Nair, V., and Hinton, G. E. 2010. Rectified linear units improve restricted Boltzmann machines. In *Proceedings of the 27th International Conference on Machine Learning (ICML-10)*, 807–814.

Pan, S.; Shavit, G.; Penas-Centeno, M.; Xu, D.-H.; Shapiro, L.; Ladner, R.; Riskin, E.; Hol, W.; and Meldrum, D. 2006. Automated classification of protein crystallization images using support vector machines with scale-invariant texture and gabor features. *Acta Crystallogr., Sect D: Biol. Crystallogr.* 62(3):271–279.

Robbins, H., and Monro, S. 1951. A stochastic approximation method. *The Annals of Mathematical Statistics* 22(3):400–407.

Sigdel, M.; Pusey, M. L.; and Aygun, R. S. 2013. Real-time protein crystallization image acquisition and classification system. *Crystal Growth & Design* 13(7):2728–2736.

Snell, E. H.; Lauricella, A. M.; Potter, S. A.; Luft, J. R.; Gulde, S. M.; Collins, R. J.; Franks, G.; Malkowski, M. G.; Cumbaa, C.; Jurisica, I.; and DeTitta, G. T. 2008a. Establishing a training set through the visual analysis of crystallization trials. part II: crystal examples. *Acta Crystallogr., Sect D: Biol. Crystallogr.* 64(11):1131–1137.

Snell, E. H.; Luft, J. R.; Potter, S. A.; Lauricella, A. M.; Gulde, S. M.; Malkowski, M. G.; Koszelak-Rosenblum, M.; Said, M. I.; Smith, J. L.; Veatch, C. K.; Collins, R. J.; Franks, G.; Thayer, M.; Cumbaa, C.; Jurisica, I.; and Detitta, G. 2008b. Establishing a training set through the visual analysis of crystallization trials. part I: 150 000 images. *Acta Crystallogr., Sect D: Biol. Crystallogr.* 64(11):1123–1130.

Szeliski, R. 2011. *Computer Vision: Algorithms and Applications*. Springer.