Hawo H. Höfer*, André Orth, Simon Schweidler, Ben Breitung, Jasmin Aghassi-Hagmann, and Markus Reischl

Quantitative Convolutional Neural Network Based Multi-Phase XRD Pattern Analysis

https://doi.org/10.1515/cdbme-2024-2075

Abstract: X-ray diffraction (XRD) is commonly used to analyze phase compositions of crystalline samples. Medical applications include the analysis of biotechnological materials and gall- and kidney stones, where composition can inform pathology assessment. XRD analysis methods like Rietveld refinement requires expert knowledge, and multi-phase sample analysis is especially challenging and time consuming. Large-scale medical and biotechnological experiments can therefore be hindered by the need to perform analysis tasks using XRD. Here, we present preliminary results on an automated convolutional neural network (CNN) based method for sample composition analysis using XRD patterns. It can aid experts' analysis using initial estimations, and enable basic judgements for non-experts. Furthermore, we confirm the intuitive notion that analysis performance degrades with sample complexity through systematic investigation using a synthetic dataset.

Keywords: XRD composition analysis, convolutional neural networks, deep learning

1 Introduction

In medicine and medical technologies, X-ray diffraction (XRD) is used to characterize crystalline biological samples (human teeth and gall-/kidney stone samples [1]), but applications extend to the analysis of bioceramics [2] and implant surface coatings [3]. X-ray diffraction methods measure interactions between radiation and the sample's crystal structure. They produce characteristic patterns containing reflections corresponding to crystal lattice planes in the sample, with multi-phase samples producing more — possibly overlapping — reflections. Analysis of these patterns using methods like Rietveld refinement can be difficult and time-consuming, impeding larger studies.

***Corresponding author: Hawo H. Höfer,** Institute for Automation and Applied Informatics (IAI), Karlsruhe Institute of Technology (KIT), Eggenstein-Leopoldshafen, Germany, e-mail: hawo.hoefer@kit.edu
André Orth, Markus Reischl, IAI, Karlsruhe Institute of Technology (KIT), Eggenstein-Leopoldshafen, Germany
Simon Schweidler, Ben Breitung, Jasmin Aghassi-Hagmann, Institute of Nanotechnology, Karlsruhe Institute of Technology (KIT), Eggenstein-Leopoldshafen, Germany

Recent advances in machine learning have enabled high-throughput experiments in medical and biological research by taking advantage of automated image processing [4, 5]. These effects can also be harnessed by development of neural network based methods for automated XRD pattern analysis. As with medical image processing, only limited quantities of labeled XRD data are available. Therefore, many methods use fully [6–9] or partially synthetic training data [10, 11].

Qualitative XRD pattern analysis deals with categorization of XRD patterns including phase identification [7–12] and more general classification tasks [6]. Dataset sizes vary greatly, ranging from less than 100 experimental samples [12] to datasets exceeding one million simulated XRD patterns [11]. Conventional machine learning methods like k-nearest-neighbors perform similarly to shallow neural networks (accuracies higher than 90%) [12] when trained on small (experimental) datasets. Larger and more complex datasets enable neural networks to perform better than conventional alternatives, achieving classification accuracies of upwards of 80% [10]. Later work reaches accuracies of 90–95% [6, 7].

Estimating phase fractions from XRD patterns (quantitative analysis) has proven more challenging. Recent literature on kidney stone analysis [12] and geological sample classification [13] suggests that conventional methods like k-nearest-neighbors regressors slightly outperform shallow neural networks when trained on small datasets. Hosein *et al.* [14] achieve similar performance to Greasley *et al.* [12] on the same dataset using an optimization-based approach. Kim *et al.* [15] train deep neural networks for composition analysis on small augmented geological datasets, but their model performance varies strongly with the analyzed mineral.

Lee *et al.* [16] employ a two-stage method of phase identification and subsequent dispatch to 1540 different models for composition analysis to achieve near perfect model performance. However, their method relies on a very large model and a dataset of more than 10^7 2–3 component mixtures of 21 phases, making training and distribution unwieldy.

Here, we present a proof of concept for automated CNN-based XRD composition analysis and a training method implemented on a subset of the Crystallography Open Database (COD) [17]. Furthermore, we examine the effects of material pool size and the number of coexisting phases per sample on model performance.

2 Method

Data Preparation

Supervised machine learning algorithms require large quantities of examples (training data) which map model inputs to desired outputs. Due to the lack of sufficient labeled data, we create a synthetic training dataset by simulating XRD patterns using the *pymatgen* software package [18].

We sampled 50 random structures from a subset of the COD [17] crafted to remove crystalline polymers from the dataset. Following previous work from our group (Schuetzke *et al.* [6, 8]), 500 strain conditions with random amplitudes less than 0.05 were applied to the materials' unit cells. Our benchmark dataset of $50 \cdot 500 = 25000$ single-phase XRD patterns was simulated from these strained unit cells.

Storing a sufficiently sized dataset of multi-phase XRD patterns is unfeasible due to the size of the configuration space. Instead, they are constructed during training by linear combination of single-phase patterns. The complexity of the constructed patterns depends on the maximum number of coexisting phases per sample K and the total number of different materials in the dataset N. Strain conditions applied to the same material count as distinct phases for the purposes of K, but are mapped to the same model output during training.

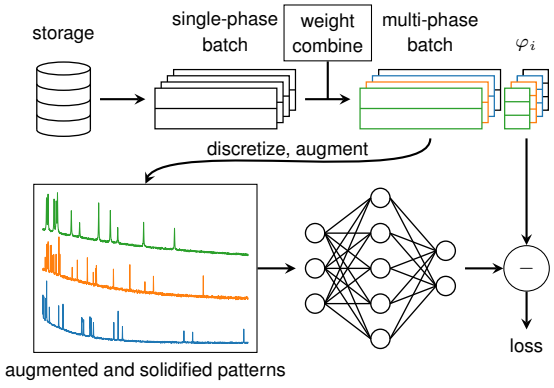

Fig. 1: The training data pipeline: single-phase XRD patterns from a batch are weighted and combined. The mixed pattern's peak positions are discretized, peak shapes are placed. Finally, the pattern is augmented using noise and background signals before being input into the model.

Figure 1 shows the training data pipeline. Batches of 512 single phase XRD patterns are drawn from a dataset of size $N \cdot 500$ during training. After combination using random phase fractions, the peak positions are discretized. Pseudo-Voigt profiles with random mixing parameters for the Gaussian and Lorentzian ($\eta \in [0.1, 0.9]$) are placed at the discretized peak positions (like Schuetzke *et al.* [6]). We determine their full width at half maximums according to the Scherrer equation, similar to Park *et al.* [19], using uniformly random crystallite sizes ($10\,\mathrm{nm}$ to $100\,\mathrm{nm}$) like Schuetzke *et al.* [6]. For simplicity, device-specific peak broadening and preferred crystal orientations are ignored. After peak placement, the patterns are normalized to the range $[0, 1]$ before Gaussian noise (3 to 7% of maximum signal magnitude) and random background signals (exponential curve) are added.

Model Architecture

As shown in Figure 2, our model consists of a convolutional (Conv-) and fully connected (FC-) stage. The convolutional stage is comprised of 3 blocks containing batch normalization, convolutional and pooling layers. After each convolutional block, copies of the data are passed through convolutional reduction layers (**R1**-**R3**) to adjust their sizes. Then, they are concatenated (similar to Yuan *et al.* [20]) and passed through the fully connected stage to produce the predicted phase fractions $\varphi_i \in [0, 1]$. In total, the proposed model architecture has about $30 \cdot 10^6$ parameters. Table 1 shows the convolutional layers' hyperparameters.

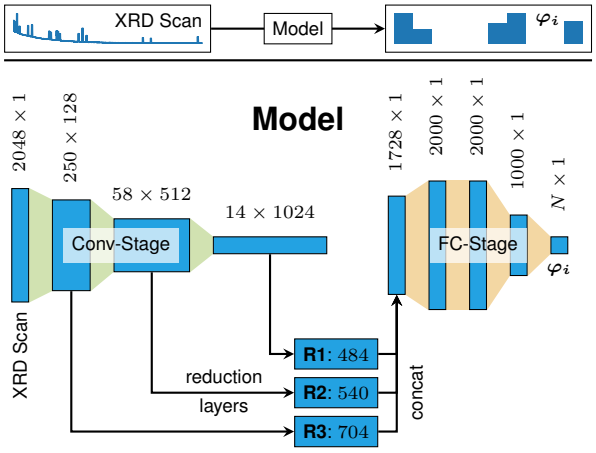

Fig. 2: The model architecture used in this paper. It is comprised of a convolutional (Conv-) and a fully connected (FC-) stage. Data shapes are represented in blue. **R1**, **R2** and **R3** indicate data shapes after passing through the respective reduction layers. Filter sizes are displayed in Table 1.

Batch normalization is applied before each layer except the first. Where specified in Table 1, convolutional layers are followed by Max pooling with a kernel size and stride of 2. All layers except the last one use leaky rectified linear unit activations with a negative slope of 0.01. In the last layer, softmax activation ensures that the outputs sum to one.

Tab. 1: Layers used in the convolutional blocks / reduction layers of the model architecture. All convolutions used a stride of 1 and no padding. **R** indicates reduction layers. Kernel size is indicated by K.

block	filters	K	pool
1	64	7	yes
	64	7	yes
	128	7	yes
R1	4	9	yes
2	256	7	yes
	512	7	yes
R2	20	5	yes
3	1024	7	yes
	1024	7	no
	1024	7	no
R3	64	4	no

Loss Function

During training, model parameters are adjusted using the ADAM optimizer [21]. This requires a loss function to evaluate model performance. Both the model prediction P and target model output T must sum to one, because they are compositions. Therefore, they can be interpreted as probability distributions. A sensible loss formulation is a distance between probability distributions, such as the total variation distance TVD (see Equation 1). The TVD can be interpreted as the average material portion of a given physical sample assigned an incorrect phase.

$$\mathrm{TVD} = \frac{1}{2}\sum_{i=1}^{N}|T_i - P_i| \quad \left(= \frac{N}{2}\mathrm{MAE}\right);$$
$$P_i, T_i \in [0.0, 1.0]; \quad \sum_{i=1}^{N} T_i = 1; \quad \sum_{i=1}^{N} P_i = 1; \quad (1)$$

Additionally, using TVD enables comparison of losses between models of different output lengths, which is especially useful when analyzing the influence of sample complexity.

3 Experiments and Results

We use a 60/40 training/validation split, and train all our models for 3000 epochs using ADAM [21] with a learning rate of $2 \cdot 10^{-4}$. This amounts to $0.6 \cdot 3000 \cdot 500 = 9 \cdot 10^5 \cdot N$ different samples per training. Our test dataset was constructed from the same 50 components as described in Section 2, but with 200 different strain conditions. They are combined with different ratios during testing. We measure model performance using the mean TVD (see Equation 1) on the test dataset ($\overline{\mathrm{TVD}}_{\mathrm{test}}$).

We explored the effects of sample complexity by varying the parameters N (material pool size) and K (coexisting phases per sample) as described in Equation 2. The number of epochs was kept constant to compare the effects of sample complexity on training after similar time scales. To account for variances in training and initialization, we trained and evaluated 3 models for each configuration (N, K). Training took between 50 ($N = 5$) and 160 min ($N = 20$) on an NVIDIA Titan RTX GPU with 24 GB of VRAM.

$$N \in \{5, 10, 15, 20\} \qquad K \in \{3, 5, 7\} \qquad (2)$$

An increase in dataset and sample complexity through N and K increases $\overline{\mathrm{TVD}}_{\mathrm{test}}$. Figure 3 shows the relationship between the $\overline{\mathrm{TVD}}_{\mathrm{test}}$ and N for different K. Model performance deteriorates by a factor of 5.4 when comparing the simplest ($K = 3, N = 5$) and most complicated cases ($K = 7, N = 20$). The training loss did not converge for $N > 10$ and $K > 5$, indicating that these configurations require more than the provided $9 \cdot 10^5 \cdot N$ samples for convergence. Sample complexity therefore not only affects model performance, but also increases computational and data requirements.

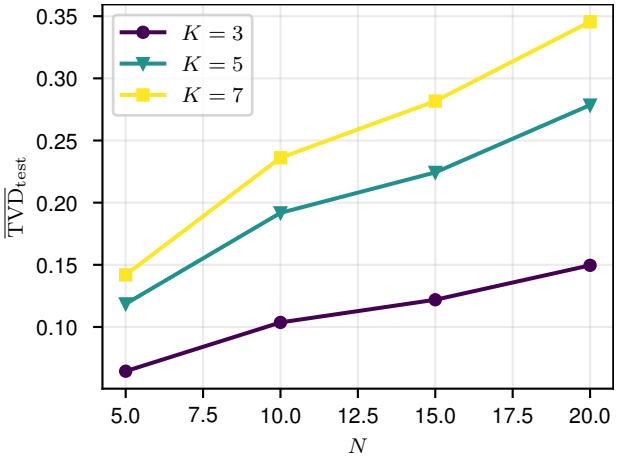

Fig. 3: Total variation distance on the test dataset $\overline{\mathrm{TVD}}_{\mathrm{test}}$ for the models trained on a dataset with given N and K. Each data point represents the mean of three runs. Except for the $(N = 5, K = 3)$-case ($\sigma \approx 0.08\mu$), standard deviations are less than 2%. All of them are therefore omitted.

Training of models for XRD pattern composition analysis is greatly influenced by the complexity of the material system. Especially if the number of coexisting phases can be assumed low, lower training times are required for acceptable precision of TVD ≈ 0.1.

Our model performs worse compared to the 400 times larger model presented in Lee *et al.* [16], which is expected given the size difference. It performs better than the models presented in Greasley *et al.* [12], reaching TVDs of 0.10

($N = 10, K = 3$) and 0.06 ($N = 5, K = 3$) in comparable test cases. Their traditional machine learning algorithms reach MAEs between 0.048 (TVD $= 0.168$) and 0.083 (TVD $= 0.291$) for $N = 7$ and $K = 2$.

4 Conclusion and Future Work

The presented model architecture and training framework constitute a possible starting point for automated XRD pattern composition analysis. Our model can aid specialists by providing initial guesses for sample compositions. With further improvements, it could facilitate automation of analysis steps needed in drug discovery or materials screening processes. While preceding publications have produced lightweight but imprecise or unwieldy but accurate models, our model delivers a tradeoff between those extremes. Downsizing and increasing parameter efficiency reduce training times and make models easier to distribute. This enables on-demand training of neural networks for problem-specific XRD analysis solutions.

While training on synthetic data generally cannot be avoided, our future work should include testing on experimental XRD patterns to validate the synthetic training data, as done in Lee *et al.* [16]. For randomly chosen material components, this data generally does not exist and testing requires the selection of a real material system more specific to the end use caes. This likely requires further tweaks to the augmentation process, such as the modeling of device-specific inaccuracies related to monochromator and collimator properties.

Additionally, effects of dataset and sample complexity on other network architectures could be examined to gain a more thorough understanding of the requirements on network architecture for XRD composition analysis.

References

[1] K Bielecka *et al.* "X-ray diffraction and elemental analysis of medical and environmental samples." In: *Acta Physica Polonica A* 125.4 (2014), p. 911.

[2] G Poralan *et al.* "X-ray diffraction and infrared spectroscopy analyses on the crystallinity of engineered biological hydroxyapatite for medical application." In: *IOP conference series: materials science and engineering*. Vol. 79. 1. IOP Publishing. 2015, p. 012028.

[3] K Banaszek *et al.* "Complex XRD and XRF characterization of TiN-TiCN-TiC surface coatings for medical applications." In: *Solid state phenomena* 225 (2015), p. 159.

[4] T Tronser *et al.* "Droplet microarray: miniaturized platform for rapid formation and high-throughput screening of embryoid bodies." In: *Lab on a Chip* 18.15 (2018), p. 2257.

[5] MP Schilling *et al.* "Grid screener: A tool for automated high-throughput screening on biochemical and biological analysis platforms." In: *IEEE access* 9 (2021), p. 166027.

[6] J Schuetzke *et al.* "Accelerating Materials Discovery: Automated Identification of Prospects from X-Ray Diffraction Data in Fast Screening Experiments." In: *Advanced Intelligent Systems* 6.3 (2024), p. 2300501.

[7] JW Lee *et al.* "A deep-learning technique for phase identification in multiphase inorganic compounds using synthetic XRD powder patterns." In: *Nature communications* 11.1 (2020), p. 86.

[8] J Schuetzke *et al.* "Enhancing deep-learning training for phase identification in powder X-ray diffractograms." In: *IUCrJ* 8.3 (2021), p. 408.

[9] NJ Szymanski *et al.* "Probabilistic deep learning approach to automate the interpretation of multi-phase diffraction spectra." In: *Chemistry of Materials* 33.11 (2021), p. 4204.

[10] F Oviedo *et al.* "Fast and interpretable classification of small X-ray diffraction datasets using data augmentation and deep neural networks." In: *npj Computational Materials* 5.1 (2019), p. 60.

[11] H Wang *et al.* "Rapid identification of X-ray diffraction patterns based on very limited data by interpretable convolutional neural networks." In: *Journal of chemical information and modeling* 60.4 (2020), p. 2004.

[12] J Greasley *et al.* "Exploring supervised machine learning for multi-phase identification and quantification from powder X-ray diffraction spectra." In: *Journal of Materials Science* 58.12 (2023), p. 5334.

[13] SY Park *et al.* "Application of machine learning to quantification of mineral composition on gas hydrate-bearing sediments, Ulleung Basin, Korea." In: *Journal of Petroleum Science and Engineering* 209 (2022), p. 109840.

[14] P Hosein *et al.* "An optimization-based supervised learning algorithm for PXRD phase fraction estimation." In: *Materials Today Communications* 36 (2023), p. 106423.

[15] D Kim *et al.* "Predicting mineralogy by integrating core and well log data using a deep neural network." In: *Journal of Petroleum Science and Engineering* 195 (2020), p. 107838.

[16] JW Lee *et al.* "A data-driven XRD analysis protocol for phase identification and phase-fraction prediction of multiphase inorganic compounds." In: *Inorganic Chemistry Frontiers* 8.10 (2021), p. 2492.

[17] S Gražulis *et al.* "Crystallography Open Database (COD): an open-access collection of crystal structures and platform for world-wide collaboration." In: *Nucleic acids research* 40.D1 (2012), pp. D420–D427.

[18] SP Ong *et al.* "Python Materials Genomics (pymatgen): A robust, open-source python library for materials analysis." In: *Computational Materials Science* 68 (2013), p. 314.

[19] WB Park *et al.* "Classification of crystal structure using a convolutional neural network." In: *IUCrJ* 4.4 (2017), p. 486.

[20] P Yuan *et al.* "HS-ResNet: Hierarchical-split block on convolutional neural network." In: *arXiv preprint arXiv:2010.07621* (2020).

[21] DP Kingma *et al.* "Adam: A method for stochastic optimization." In: *arXiv preprint arXiv:1412.6980* (2014).