

Determining crystallographic orientation via hybrid convolutional neural network

Zihao Ding, Chaoyi Zhu^{*}, Marc De Graef

Department of Materials Science and Engineering, Carnegie Mellon University, Pittsburgh, PA 15213, USA

ARTICLE INFO

Keywords:

Electron backscatter diffraction
Convolutional neural network
Dictionary indexing
Hybrid indexing

ABSTRACT

A recent paradigm shift in the electron diffraction community has benefited from accessibility of large data sets and ever more complex designs of convolutional neural networks (CNNs). However, this shift from conventional feature engineering to analyzing high-level features extracted from CNN is often accompanied by a reduction in accuracy and sensitivity. Particularly, CNN based crystal orientation indexing using electron backscatter diffraction is sensitive to noise, reducing the overall accuracy. In this study, a new hybrid indexing approach has been developed to integrate dictionary indexing (DI) with a trained CNN to achieve extraordinary speed and robustness against noise simultaneously.

1. Introduction

Arguably, one of the most exciting moments during the history of materials science is the realization that bulk crystalline materials are composed of smaller, differently oriented crystals. Since the orientation of each crystal determines how it will react differently to the same external stimuli, constitutive laws accurately describing a macroscopic property of material needs to incorporate the ensemble of crystallographic orientations [1]. Spatially resolved crystallographic information to interpret and model material properties is often determined by diffraction-based techniques such as high-energy X-ray diffraction microscopy (HEDM) [2–4] and electron backscatter diffraction (EBSD) [5]. Unlike electron diffraction, HEDM, carried out exclusively at beamlines, provides a 3D structure information. Nevertheless, EBSD has become increasingly popular because of its accessibility as part of a standard university-level research laboratory with a scanning electron microscope (SEM). It offers advantages in spatial/angular resolution (40 nm [6]/0.5° [7]) and fast sampling of crystallographic/microstructural information over a large area of interest.

Early development of EBSD can be attributed to multiple research groups around the world [8–10]. Its modern popularity stem from the development of CCD/CMOS imaging sensors and fully automated computer-based data analytics [11,12]. More specifically, the implementation of computer vision technique i.e. Hough/Radon transform [13] has replaced the tedious manual indexing. In the context of crystallographic orientation determination, the conventional Hough-

transform indexing (HI) approach transforms the spatial coordinates of Kikuchi bands into Hough space (distance-angle space) and then calculates at least three inter-planar angles to be compared with a lookup table (LUT) [11,12]. Besides advanced hardware with high sensitivity, the precision and accuracy of EBSD orientation measurement relies strongly on the indexing method to extract the location of the Kikuchi bands from electron backscatter diffraction patterns (EBSPs) [14,15]. Of course, the accuracy of orientation measurement with EBSD also depends on the accuracy of the geometric setup of the measurement e.g. sample tilt, detector tilt, pattern center [16]. This dimensionality reduction technique i.e. Kikuchi bands to points in the Hough space inevitably results in loss of information. While HI is computationally efficient, it is very dependent on the signal-to-noise ratio and almost impossible to distinguish pseudosymmetries [17,18].

Instead of extracting specific diffraction features, the dictionary indexing (DI) [19], or more recently the spherical indexing (SI) [20,21], exploits the use of simulated dynamical electron diffraction patterns in order to take the full diffraction pattern into consideration [22]. Subsequently, a search algorithm can be efficiently built to traverse through the similarity landscape between a dictionary of simulated patterns and experimental patterns to obtain the correct Euler angle triplets. These pattern matching methods are robust against noise and even sensitive to the presence of subtle differences due to pseudosymmetries [18,20,23]. In some cases, researchers can even use pattern matching methods to infer c/a ratio of martensite [24,25] and the full deformation tensor [26]. For DI method, the size of the dictionary is enormous (on the order

^{*} Corresponding author.

E-mail addresses: zihao@andrew.cmu.edu (Z. Ding), zchaoyi@andrew.cmu.edu (C. Zhu), degrae@cmu.edu (M. De Graef).

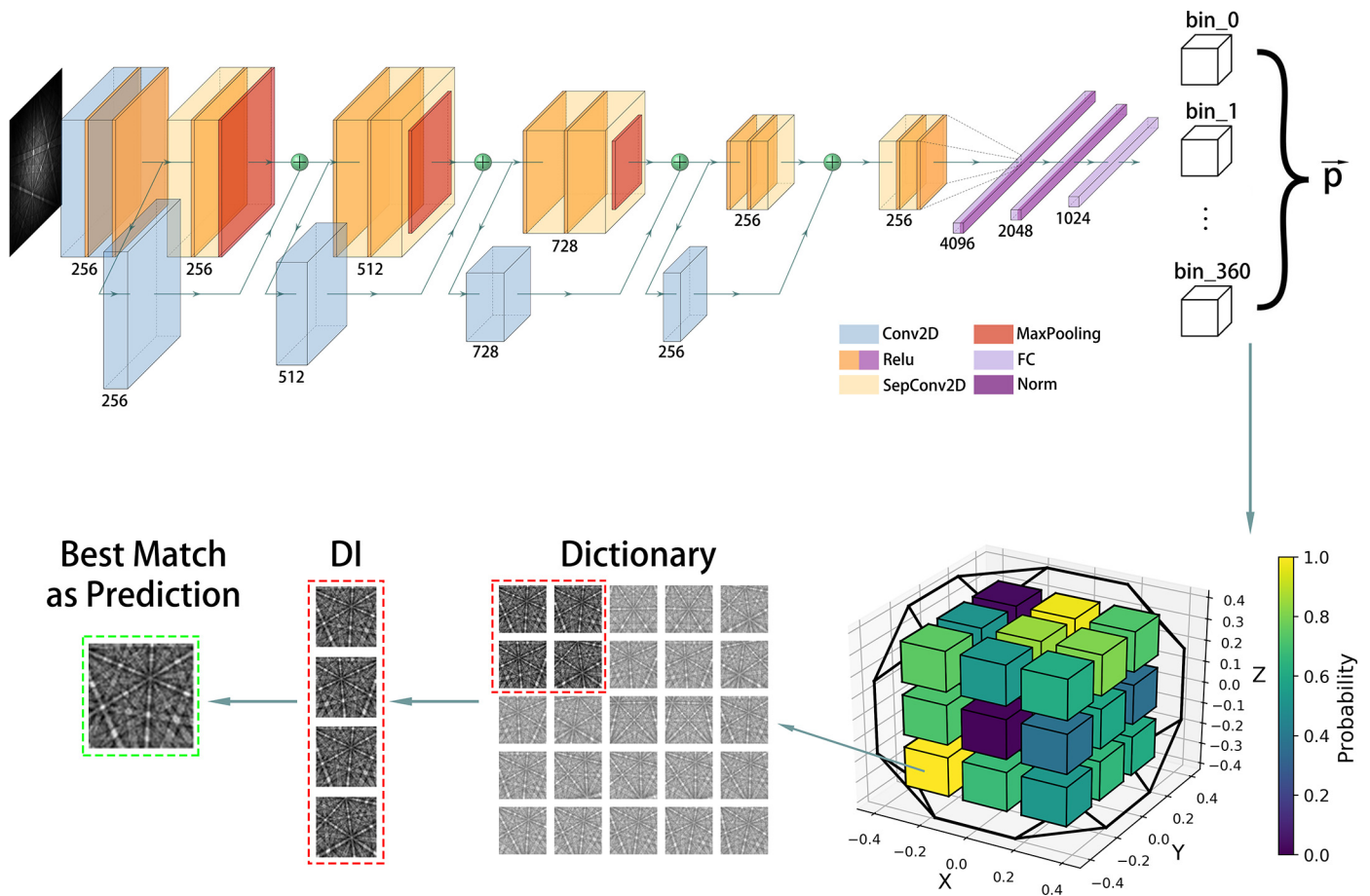


Fig. 1. Schematic of the EBSDI-CNN architecture.

of 10^5 patterns per dictionary) in order to uniformly sample orientation space in the cubochoric representation [27]. Its root cause of the computational complexity is associated with the fact that each experimental pattern needs to be compared with all the dictionary patterns (i. e. an enumeration algorithm), making this process very computationally demanding. Thus, an alternative method that improves the speed of DI is needed while preserving its sensitivity to noise.

With exponential growth in computing power, the material characterization community is also transitioning to be more data-driven due to the advancement in data collection and integrating machine learning algorithms to accelerate the data interpretation [28]. For instance, Holm et al. [29] have trained a CNN instance segmentation system based on Fast R-CNN [30] to identify gas atomized metal powders from SEM micrographs, a task that was typically approached manually and subjectively. Moreover, DeCost et al. [31] proposed another CNN model to semantically segment complex microstructures (including grain boundary carbide, spheroidized particle matrix and Widmanstätten cementite) in ultra-high carbon steel. In the electron diffraction community, earliest application of machine learning technique in EBSD is the use of artificial neural network to verify the detected bands in Radon space [32]. More recently, CNNs with various output spaces have been previously designed to classify phases [33], and space groups [34–36]. Determination of crystallographic orientation using machine learning algorithms has also been explored by several search groups [37–39].

Due to the continuous nature of orientation space, crystallographic orientation determination is essentially a regression problem and therefore poses a challenge for CNN. Unlike a classic image classification problem that uses normalized categorical values for a given number of categories, the recent development by our group uses an end-to-end CNN (EBSD-CNN) to determine the unit quaternion representing the

corresponding orientation [39]. However, two critical issues remain unsolved in the EBSD-CNN model are: [1] the training of the model for a regression problem is computationally demanding; [2] the effect of noise on the orientation accuracy. In this study, we present a new hybrid design strategy for CNN based model to determine crystallographic orientation. The seamless integration of CNN and DI (EBSDI-CNN) allows the model to be trained with less effort and at the same time achieve unparalleled speed, accuracy and sensitivity.

2. Methodology

2.1. EBSDI-CNN design

As mentioned in the introduction, EBSDI-CNN is a serialized workflow with a classification CNN and a DI indexing framework as the main components. A flowchart is shown in Fig. 1. First, a high energy electron beam (20 kV) is used to probe a tilted crystalline sample surface. Upon interaction, elastically and inelastically scattered electrons emerge from the surface towards a detector equipped with a phosphor screen and a CCD or CMOS based camera at the back. Collected raw patterns are usually pre-processed to remove the background intensity and optics distortion. For HI, detailed description of how to transform orientation from the detector screen to Cartesian crystal frame is described elsewhere [40]. In this study, an additional pattern pre-processing module, also available as part of the open-source project PyEMEBSDI,¹ is used to remove artifacts and improve pattern quality of experimental patterns [41].

¹ https://github.com/Darkhunter9/PyEMEBSDI_wrapper

As shown in Fig. 1, the design of CNN is based on Xception [42] from Google with minor simplifications and customization. The entry flow consists of two 2D convolutional layers and rectified linear units (ReLU) as activation function. In the middle flow, the conventional convolutional layers in the are replaced with depth-wise separable convolutional layers to decrease the number of parameters, lowering the computational complexity. Also, max-pooling units are added to reduce the dimensionality and further reduces the computational cost. In the exist flow, the extracted high-level features are fed into fully connected layers after two more rounds of convolution.

In contrast to our previous EBSD-CNN approach, the CNN part in the hybrid method is used to predict several sub-spaces in the entire orientation space to which the pattern might belong. In this way, the subsequent DI step no longer needs to traverse through the whole dictionary. Thus, the CNN output corresponds to the probability of a pattern belonging to a specific bin in the segmented orientation space. In this study, we use a total of 361 orientation bins generated from the EMsampleRFZ function [27,43] provided by the open-source EMsoft² package. The loss function is switched from disorientation/misorientation used in the orientation space to the sparse categorical cross-entropy [44]:

$$L = -\frac{1}{361} \sum_{i=1}^{361} y_i \log(\hat{y}_i), \quad (1)$$

where y_i is the ground truth and $\hat{y}_i$ is the prediction.

The link between the CNN and the DI algorithms is the dictionary generator that customizes the exact number of sub-dictionaries for each pattern to be categorized into based on the corresponding CNN prediction. The resultant dictionary can be readily obtained by grouping dictionary patterns in the first K number of bins with high probabilities (Top_K). Next, these dictionary patterns are converted into normalized column vectors. Using a similarity metric such as the normalized dot product (NDP), every experimental pattern (normalized column vector) can be compared against all dictionary patterns and ranked according to the NDP values. Finally, the orientation of an experimental pattern is obtained by the weighted average quaternions of top M matches (the top M number of dictionary patterns ranked according to the NDP values) [14].

Since the CNN part is realized using Python and Tensorflow, we have developed Python wrappers for the DI module from EMsoft, originally written in Fortran 90. Through C++ extensions in Python, the wrapper routines provide great run-time efficiency as well as native support for multi-GPU acceleration PyEMEBSDDI.

2.2. Data preparation and partitioned training

As the CNN aims to predict approximate orientation bins for a given EBSD pattern, the first task is to discretize orientation space into an array of sub-spaces (bins). The label of each bin is the orientation at the bin center, in the form of a unit quaternion:

$$\ell_i = (l_i, \mathbf{l}_i), \quad (2)$$

where l_i is the scalar part of the quaternion for bin $i \in [1 \dots N_b]$ (N_b is the number of bins), and $\mathbf{l}_i$ the vector part. The training data obtained from EMsoft consists of pairs of simulated patterns and their unit orientation quaternions s_j ($j \in [1 \dots N_t]$ with N_t the number of training patterns):

$$s_j = (o_j, \mathbf{o}_j) \quad (3)$$

All patterns falling into a particular bin will share their label; thus, the bin label i of each training pattern j is determined by finding the bin whose label i shows a minimum disorientation angle with respect to the

orientation of the training pattern (taking into account the rotational point group symmetry with quaternion operators $\mathcal{C}_k$, $k \in [1 \dots N]$ with N the order of the point group):

$$i = \underset{i,k \in [1 \dots N]}{\operatorname{argmin}} \omega_{k,i,j} \equiv \underset{i,k \in [1 \dots N]}{\operatorname{argmin}} 2 \arccos[(\mathcal{C}_k \ell_i) \cdot (\mathcal{C}_i s_j)] \quad (4)$$

The dot product in this relation is the standard dot product between two quaternions, i.e., $\ell_i \cdot s_j = l_i o_j + \mathbf{l}_i \cdot \mathbf{o}_j$. As stated before, in this study we use a total of $N_b = 361$ bins. This is an adjustable parameter that balances the accuracy and acceleration effect of the CNN. In this case, it is likely that the prediction of the label of an experimental pattern will be incorrect. Nevertheless, the size of the sub-dictionary for each bin will be smaller, leading to an improved acceleration.

Since the CNN is the only part of the algorithm that requires supervised learning and the DI algorithm is not involved in the data preparation, the subsequent indexing result of DI following CNN will always be the same as that obtained by traversing the complete dictionary given a correct prediction from the CNN.

2.3. Analysis of the trained system

The material used for training, validation and testing (simulated patterns) in this study is pure Nickel ($a = b = c = 352.4$ pm, $\alpha = \beta = \gamma = 90^\circ$) with a face-center cubic (FCC) crystal structure. To assess the robustness against the noise of the hybrid indexing method, a specific area of Inconel 600 nickel alloy was scanned multiple times using various exposure times (from 0.17 ms to 8.03 ms) but constant total exposure, resulting in a series of data sets with different noise levels [45]. In this study, three metrics about the trained system are evaluated:

1. Hit rate of the CNN under different noise levels using different Top_K values;
2. Indexing accuracy of the whole system compared with pure DI;
3. Indexing speed of the whole system compared with pure DI. The time consumed by the CNN and DI parts are recorded separately.

Since CNN used in the framework is for classification, the hit rate can be quantitatively assessed by metrics consisting of accuracy, precision, recall and F1 score [46]:

$$\text{Accuracy} = \frac{TP + TN}{TP + TN + FP + FN}, \quad (5)$$

$$\text{Precision} = \frac{TP}{TP + FP}, \quad (6)$$

$$\text{Recall} = \frac{TP}{TP + FN}, \quad (7)$$

$$F1 - \text{Score} = 2 \times \frac{\text{Recall} \times \text{Precision}}{\text{Recall} + \text{Precision}}, \quad (8)$$

where TP, TN, FP, FN are true positive, true negative, false positive and false negative, respectively.

For problems with multiple classes, the metrics are computed for each class in the dataset and then aggregated in a micro way:

$$\text{Micro} : B \left(\sum_i TP_i, \sum_i FP_i, \sum_i TN_i, \sum_i FN_i \right), \quad (9)$$

where $i \in [1 \dots q]$ (q is the number of classes in the data set), and B is the metric.

2.4. Analysis of the acceleration

The acceleration of the EBSDDI-CNN is related to two factors: 1) the number N_b of orientation bins used in the CNN part, and 2) the texture in the EBSD data set. The first contributing factor is easy to explain: the

² <https://github.com/EMsoft-org/EMsoft>

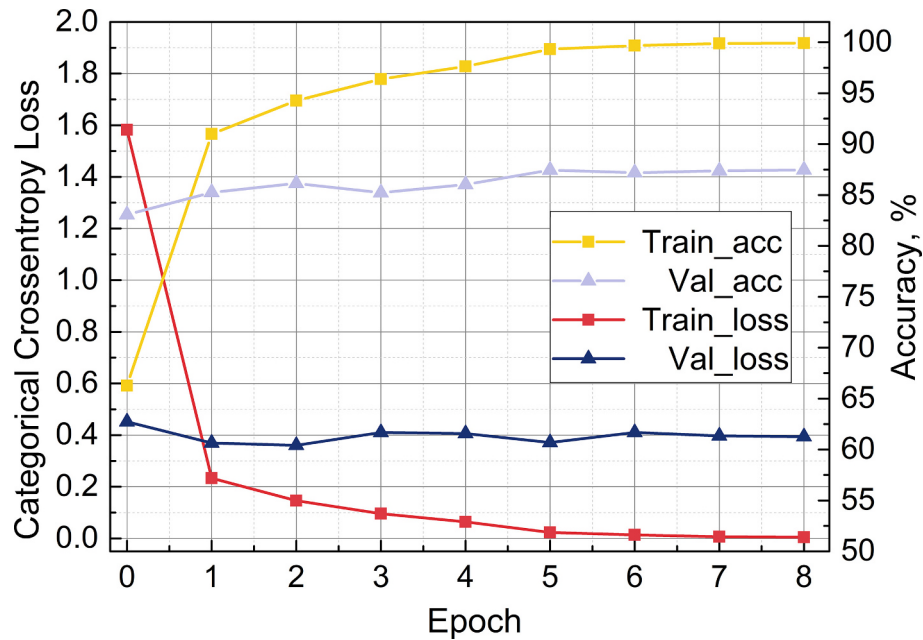


Fig. 2. Training History of EBSDI-CNN method including training accuracy (yellow), validation accuracy (light purple), training loss (red), and validation loss (blue). The training of EBSDI-CNN converges with a few epochs, much faster than EBSD-CNN. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

more bins are used in the CNN, the smaller the number of patterns per bin to traverse during the DI step. The second contributing factor is, in fact, a direct consequence of the first and the use of equal-volume mapping in a uniform orientation sample.

An indispensable part of DI is the generation of an orientation dictionary based on a uniform sampling of orientation space using a uniform and refinable grid. The sampling method is described in detail by Roşca et al. [27]. We start from a uniform cubical grid inside a 3D cube $C \in \mathbb{R}^3$. This uniform grid with equal volume voxels is mapped onto a 3D ball $B^3 \in \mathbb{R}^3$ with an equal-volume mapping. Through an inverse Lambert type equal-volume mapping, the orientations in B^3 can be

uniformly mapped onto the Northern hemisphere of the quaternion hypersphere $S_+^3 \in \mathbb{R}^4$, which is itself isomorphic with $SO(3)$. In the presence of crystallographic symmetry, only a compact subset, i.e., the *fundamental zone*, of the total $SO(3)$ group is considered which contains rotations that are unique under a particular rotational group.

Due to topological differences, a “uniform grid” only means that the volume of the voxels during the mapping is preserved whereas the angular distance or disorientations between nearest neighbors is not. As shown by Singh et al. [43], the average disorientation of uniformly sampled orientation is a function of semi-edge length. Additionally, this means that the number of dictionary orientations grouped into the

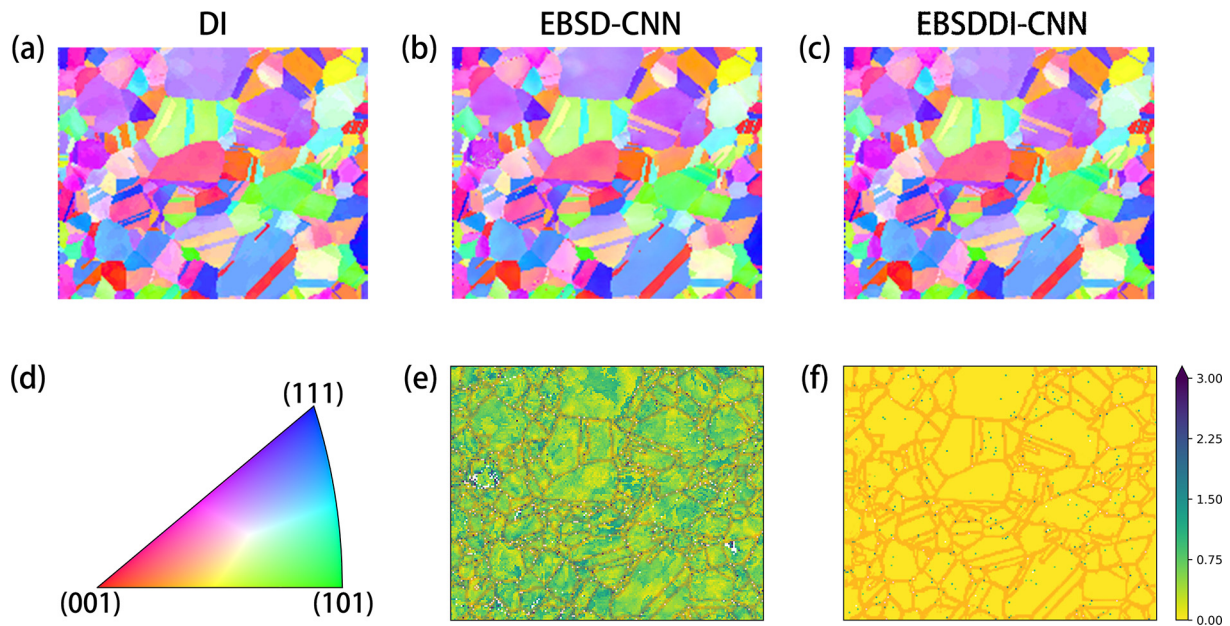


Fig. 3. IPF maps on experimental EBSD patterns: (a) Pure DI, (b) EBSD-CNN, (c) EBSDI-CNN, (d) Colour key for FCC Fundamental Zone, (e) Disorientation ($^{\circ}$) between pure DI and EBSD-CNN, and (f) Disorientation ($^{\circ}$) between pure DI and EBSDI-CNN. Grain boundaries are highlighted in red in (e) and (f). (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

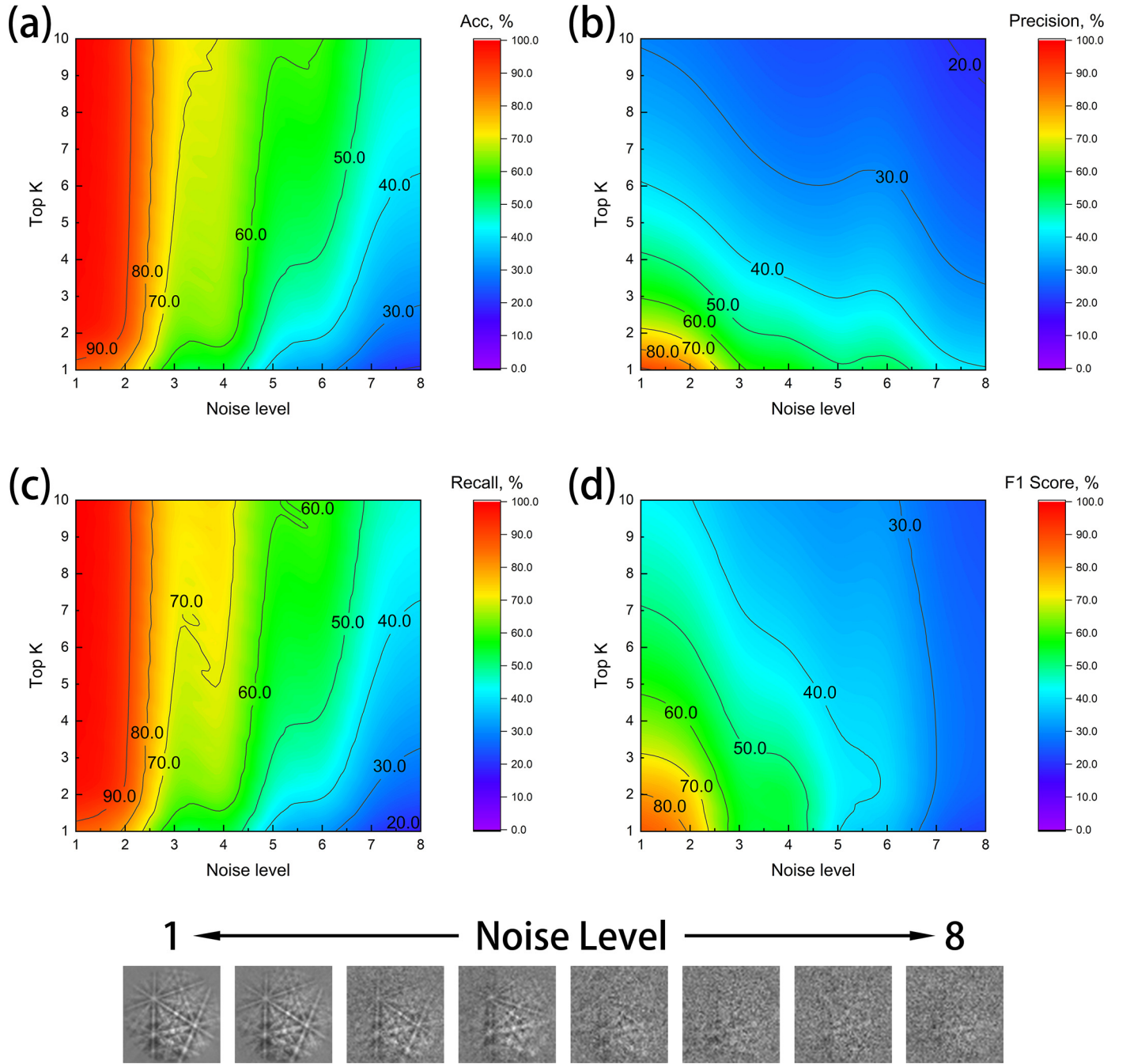


Fig. 4. Evaluation metrics of CNN with tunable Top_K value under different noise levels: (a) accuracy, (b) precision, (c) recall, and (d) F1-score.

orientation bins based on smallest disorientation values will differ significantly from bin to bin, as shown later in the discussion section.

To approximate the number of dictionary patterns per bin for any orientation, the method of spherical harmonics approximation is used to translate these discrete values located on a sphere:

$$s(r) = \sum_{m=0}^M \sum_{l=-m}^m \hat{f}_{m,l} Y_{m,l}(r), \quad (10)$$

where r is the coordinate of a bin, m and l are the degree and order of spherical harmonics $Y_{m,l}$, M is the cutoff value or the bandwidth to which degree m is summed over, $\hat{f}_{m,l}$ is the Fourier coefficient, and $s(r)$ is the spherical function approximation of the bin's values. More

specifically, we seek Fourier-coefficients $\hat{\mathbf{f}} = (\hat{f}_{m,l}, \dots, \hat{f}_{M,M})^T$ so that

the approximate spherical function minimizes the difference between the exact bin values and the approximate bin values for a given minimization strategy such as the least squares method:

$$\min_{r_n \in SO(3)} \sum_{n=1}^{N_b} |S(r_n) - s(r_n)|^2, \quad (11)$$

for bins r_n ; $S(r_n)$ is the exact number of patterns per bin, and $s(r_n)$ is the approximated number of patterns per bin evaluated from the spherical function.

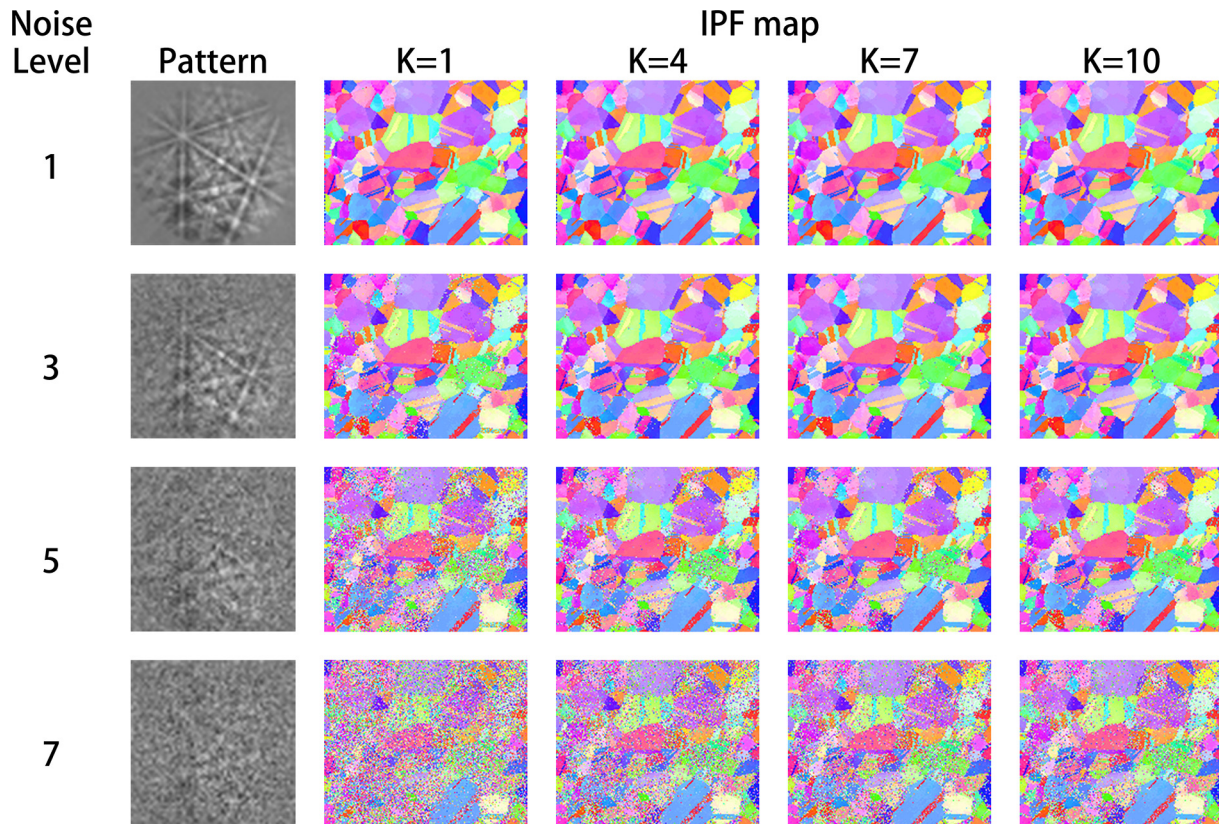


Fig. 5. IPF of the nickel experimental dataset with tunable Top_K value under different noise levels. The colour key is the same as Fig. 4(d). With $Top_K = 1$, EBSDDI-CNN has already shown a noise resistance comparable to HI. And as Top_K increases, the performance gets closer to DI.

3. Results

3.1. Training of CNN

The training history of the CNN part is shown in Fig. 2. Compared with the EBSD-CNN [39], the most significant difference is the faster convergence. The validation accuracy ($Top_K = 1$) reached 85% and became stable after only 5 training epochs, while it took over 15 epochs for the EBSD-CNN to show a competitive performance. This indicates the possibility of realizing “few shot learning” [47], which may fundamentally address the lack of versatility in a trained model and promote its wide application in practical EBSD indexing on various materials.

To confirm that the training step is effective, a set of experimental patterns with relatively good pattern quality were indexed individually by pure DI, EBSD-CNN and EBSDDI-CNN. The IPF maps based on the indexing results of three methods are shown in Fig. 3. It can be seen qualitatively that with a good hit rate in the CNN part, the result of hybrid indexing system maintains a high level of consistency with respect to pure DI.

Another obvious improvement of EBSDDI-CNN over EBSD-CNN is the indexing accuracy of points on the grain boundaries. While all training patterns are generated with an explicit orientation and a stress free state, the situation on grain boundaries is much more complicated because of the superposition of patterns from multiple grains with diverse orientations and the potential introduction of strain [48–50]. Since such patterns require a more sophisticated simulation approach and were thus not included in the training data set, it is no surprise that EBSD-CNN performed poorly near grain boundaries. In EBSDDI-CNN on the other hand, the performance near grain boundaries is improved due to the high indexing robustness of DI and the subsequent orientation refinement algorithm [51,52].

3.2. Prediction accuracy of CNN

Since DI, without any trainable parameters, has a closed form solution for any EBSP if given the dictionary, and a detailed error analysis of its orientation prediction is already available [15], it is reasonable to focus on the CNN part when evaluating the prediction accuracy of the system.

Among all metrics mentioned in Section 2.3, accuracy (also known as categorical accuracy) is the one that directly shows whether the right sub-dictionary is chosen. In most image classification problems and competitions (such as ILSVRC), to fully assess the potential of the model and avoid being penalized for objects missed in the ground truth, usually and accuracy of Top_5 choices is calculated in addition to the one with highest probability. Compared to this algorithm, the introduction of a tunable Top_K parameter here is more meaningful and practical, as all sub-dictionaries with probabilities among the classifier's top K guesses will be passed to the DI algorithm.

Fig. 4a shows the accuracy with Top_K from 1 to 10 under 8 different noise levels. While on patterns with relatively high quality, a low Top_K is enough to guarantee a good indexing accuracy, on noisy patterns, increasing Top_K value can effectively make up for CNN's sensitivity to perturbations. Even on patterns where Kikuchi band features are hard to identify, the CNN can still maintain an accuracy around 50%, using predictions with a size less than 3% of all bins.

Precision defines the fraction of correct predictions, while recall indicates the proportion of correctly identified true positives. The F1-score is the weighted average of precision and recall, taking both false positives and false negatives into account; this handles the situation of an uneven class distribution. Three metrics under different noise levels and Top_K values are plotted separately in Fig. 4b–d. It is obvious that all metrics are lowered under higher noise level. As each pattern belongs to only one class, a higher Top_K value leads to predictions with a larger

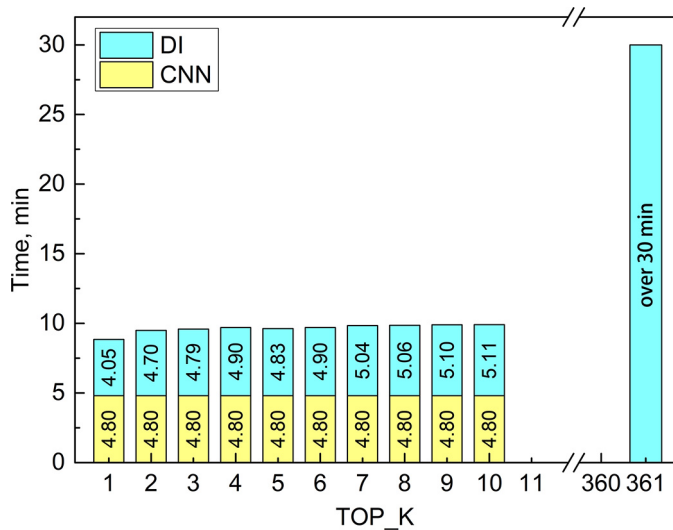


Fig. 6. Indexing speed with Top_K from 1 to 10 for the CNN and DI parts in the EBSDDI-CNN model. Yellow bars represent time taken for the trained CNN model prediction of sub-dictionaries and blue bars represent the time taken by DI. $Top_K = 361$ is equivalent to the use of DI only. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

size, causing more “false positive” counts and lowering the precision. The recall increases with Top_K but escalation is much more stable under high noise level than lower ones which already have a good hit rate. Balancing the precision and recall, the F1-score reveals that for patterns with good quality we should always stick to a lower Top_K value, while for noisy patterns, there is a certain optimal “sweet” spot. An automated workflow to determine the point of optimal balance will be discussed in the following section.

3.3. Robustness against noise

The last part of the indexing accuracy analysis is a visual check of the inverse pole figure (IPF) output by the whole system, as presented in Fig. 5. To highlight the effect of noise and Top_K 4 noise levels and Top_K values are shown. At low level noise, a small Top_K value is sufficient to obtain almost the same indexing result as obtained with DI, while reducing the amount of computation by a large fraction. At a high noise level, the robustness of the system is guaranteed by a combination of a tunable Top_K value and the intrinsic noise resistance of DI, which is much better and more flexible than the end-to-end EBSD-CNN. As an extreme case, the system can show a robustness against noise with the same upper limit of DI, by setting Top_K equal to the number of sub-

dictionaries.

Compared with EBSD-CNN, another interesting phenomenon is what happens when the indexing fails. In EBSD-CNN, most noisy patterns that cannot be indexed properly are labeled as “zero rotation” (i.e., the identity quaternion [1,0,0,0]). This is because a high noise level makes the pattern close to random input, which typically does not produce a high activation in the convolutional layers. The reaction of EBSDDI-CNN to excessively noisy patterns, on the other hand, is more similar to that of DI, assigning a random orientation in the orientation space.

3.4. Indexing speed

In addition to the indexing accuracy, acceleration is also one of this study’s original intentions. On the one hand we want to determine whether or not the indexing speed is faster than that of standard DI, i.e., whether the time consumed by the CNN step is more than that recovered by the resulting smaller dictionary. On the other hand, as described in Section 3.2, the drop in precision with a larger Top_K value indicates the increase in ratio between the computation amount and the improvement on accuracy. Thus, the increase in execution time of a larger Top_K value deserves further investigation.

The time spent on indexing the whole experimental data set with different Top_K values is shown in Fig. 6. For pure DI, it takes just over 30 min to complete the indexing. Since the model structure, as well as the output size, is independent of Top_K the execution time of the CNN step is a constant 4.8 min, using the same GPU model. The acceleration effect brought to DI part, however, is remarkable. At $Top_K = 1$, the DI step takes 4.05 min to complete the indexing of all sub-dictionaries, which is only 13.3% of the computation time of the pure DI algorithm. Even at $Top_K = 10$, the percentage only rises up to 17.0%. It is clear that the increase in computation time of the DI step is actually not proportional to Top_K . The reason lies in the design of GPU for parallel computation. With much more compute units (streaming multiprocessor in Nvidia) than CPU, the performance bottleneck of the GPU is usually the latency (instead of instruction bandwidth), meaning that if the parallelism exposed is not sufficient, the actual performance of GPU will be far from the maximum [53,54]. According to the predictions from the CNN step, patterns are divided into groups and indexed using different sub-dictionaries in the DI step. Limited experimental patterns in each batch and candidates in the corresponding sub-dictionary means that many threads remain idle. Thus, when a larger Top_K value is used, the increase of the number of patterns for each sub-dictionary does not necessarily lead to a longer indexing time.

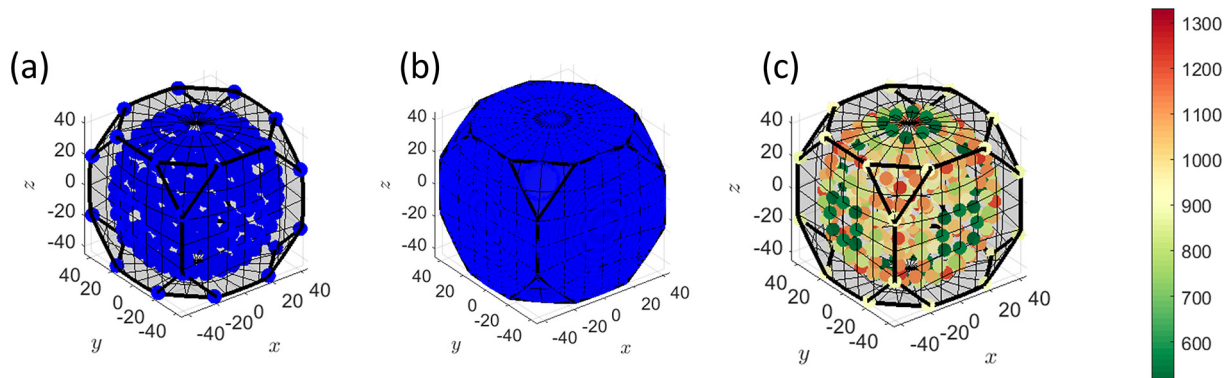


Fig. 7. Cubic Rodrigues fundamental zone for (a) orientation bins ($n = 10$) and (b) orientation dictionary ($n = 100$). (c) The number of unique orientations from the dictionary in each orientation bin.

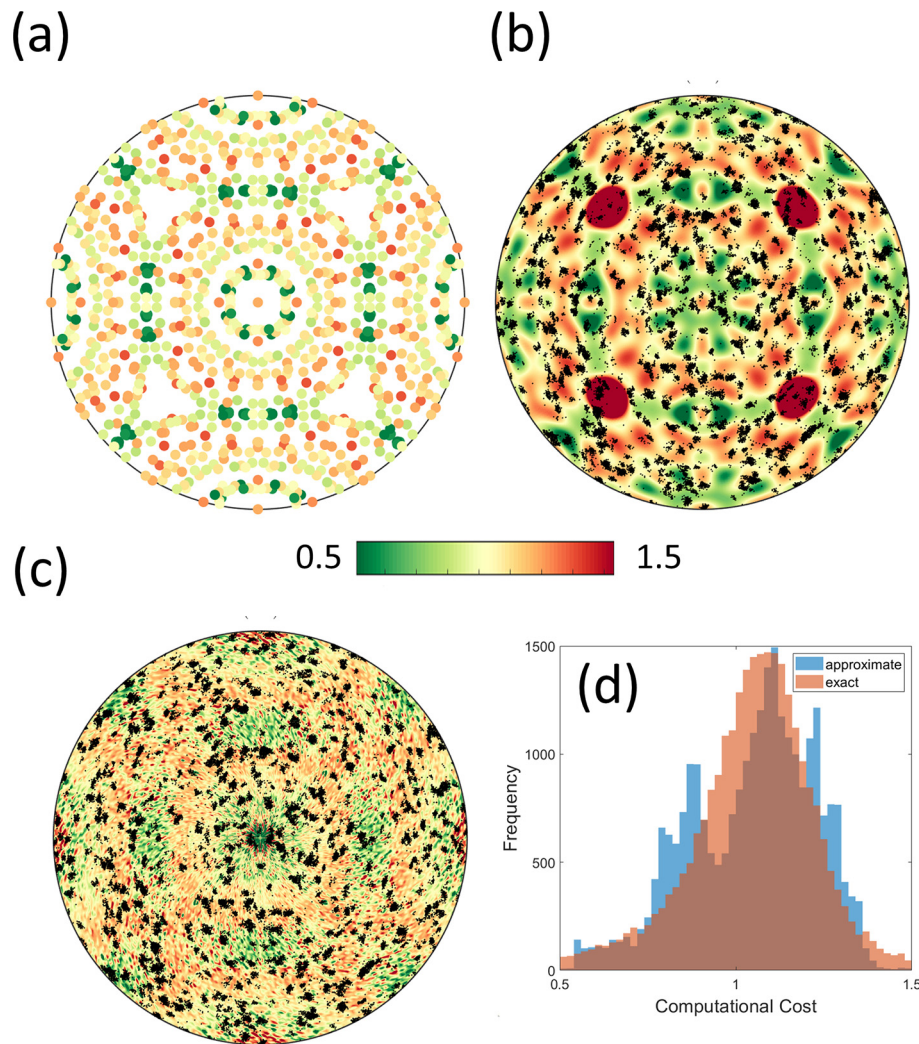


Fig. 8. (a) Orientation dependent computational cost for $n = 10$; Ni experimental data (black dots) overlaid on top of (b) the approximation of (a) with spherical harmonics (bandwidth = 33) and (c) the approximation of the computational cost for the entire orientation space with spherical harmonics (bandwidth = 1000); (d) histograms of the computational cost for Ni data set evaluated from the spherical harmonics in (b: approximate) and (c: exact).

4. Discussion

4.1. Acceleration effect analysis

For a semi-edge length N of the cubochoric grid, the corresponding total number of orientation in $SO(3)$ is simply equal to the number of uniformly sampled orientations in the cube $8N^3$. Taking into account the symmetry of the cubic system, the *fundamental zone* contains approximately $8N^3/24$ orientations. This *fundamental zone* can be visualized in Rodrigues space, as shown in Fig. 7, in which the direction is the rotation axis and the distance from the origin is the tangent of half the rotation angle. Fig. 7(c) shows how many dictionary patterns generated with $N = 100$ will be grouped into each bin.

Since a stereographic projection only preserves angular distances, orientations will not uniformly distribute on the pole figure as shown in Fig. 8(a). The value for each orientation corresponds to the ratio between the number of dictionary patterns in each bin ($N = 10$), same as data shown in Fig. 7(c), divided by the hypothetical case if dictionary patterns were to uniformly distribute among 361 bins (923 patterns per bin). Since the indexing speed of the DI step will scale inversely with respect to the number of patterns in the bin, the red dots will require 1.5 times longer time to compute compared with the hypothetical case and green dots only require half of the computational time. In other words, the value of the dots correlates directly to the computational cost of the

DI step.

However, Fig. 8(a) only represents a discrete view of the acceleration speed over the orientation space. To approximate discrete data on a sphere, a series of spherical harmonics is used to approximate the values in Fig. 8(a), which is illustrated in Fig. 8(b). Alternatively, the orientation space (using the entire dictionary) can be divided into sections belonging to a particular bin and each section has the same value of computational cost. With a higher bandwidth, the harmonic approximation of the computational cost over the entire orientation space is shown in Fig. 8(c).

By evaluating the cost of an orientation from the spherical harmonics, it is possible to statistically analyze the computational cost of a particular EBSD data set for the DI step of the EBSDI-CNN method as shown in Fig. 8(d). The “approximate” histogram uses the $N = 10$ case for the target function values to be represented by a spherical harmonic and the “exact” method uses all the dictionary orientations ($N = 100$). A higher bandwidth is necessary for the “exact” method to accurately represent the sharp changes of cost values from one bin to another. For a sample with randomly orientated grains or weak to medium texture, the overall cost usually remains close to unity. However, a strong fibre texture, for instance, close to the Euler angle triplets $[0,0,0]$ would yield a much faster computation due to the lower number of dictionary patterns close to these bins.

More recently, other methods of uniform sampling have been

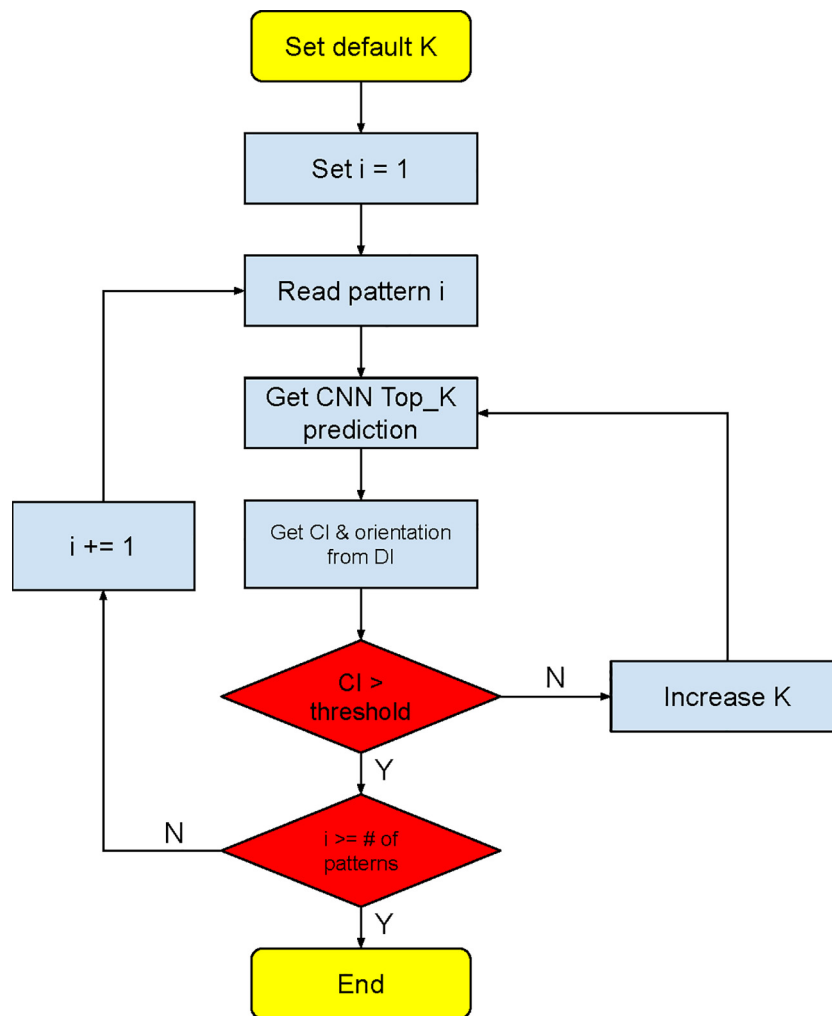


Fig. 9. Workflow for the adaptive selection of *Top_K*. The *Top_K* value is dynamically adjusted according to the confidence index returned by DI step.

proposed by Larsen & Schmidt [55] and Quey et al. [56], which might provide a more uniform distribution of the sub-dictionary sizes. The benefit of having a more uniform distribution is an easier prediction of the acceleration, regardless of the texture present in the sample, but the overall acceleration will still be comparable to that reported in the present study.

4.2. Transfer learning and few-shot training

Currently, a major problem limiting CNN's broad application in practical EBSD indexing is the lack of versatility in a trained model. In other words, a model trained on one material may not show good performance on other materials with different symmetry groups and lattice constants. Compared with EBSD-CNN, the number of training epochs required to converge the EBSDI-CNN has decreased by 60%; however, this is still a fraction of the indexing cost of a single data set.

One approach to further accelerate the training process is to use transfer learning and few-shot training. The former refers to deploying weights from models trained on other data sets in the convolutional portion, assuming the model can be improved from one domain by transferring information from a related domain [57]. The latter is the training followed by fine tuning of parameters which rapidly generalizes to the new data set containing only limited samples with supervised information [58]. This combination is originally intended for real-world machine learning scenarios where training data is expensive or difficult to collect. In our case, it is mainly used to shorten the training time,

without sacrificing much accuracy. The parameters can either be from models pre-trained on EBSD patterns of other materials within the same symmetry group, or, perhaps more challenging, from models pre-trained on image data sets in unrelated fields, such as ImageNet [59] and CIFAR100 [60].

4.3. Optimization of loss function

In this study, we used cross-entropy as the loss function, as it is one of the most commonly used loss functions in image classification problems and has been implemented in various machine learning libraries. However, this assumes that different image classes are completely irrelevant and can be mapped to a linear space, which is contrary to the prior knowledge of EBSD. Since the “distance” between any two orientations can be quantified through the disorientation, it is reasonable to cast the penalty of a categorical prediction (the bin) based on its “distance” from the ground truth in orientation space.

Several difficulties arise in the selection of an appropriate loss function:

1. An algorithm is needed to map disorientation between orientations to “distance” between bins. If the labels of each bin (i.e., the orientations at the center of each sub-dictionary) are used, the accuracy of the loss function will be dependent on the sampling density;
2. Each of the rotational point groups has a unique set of symmetry operations, making the expression of a general loss function

complex. In addition, although functions like *reduce_max* are provided in TensorFlow and PyTorch, for rotational groups of high order (e.g., 24 for point group 432), the comparison between all symmetrically equivalent orientations will remain a computationally heavy task.

4.4. Adaptive selection of hyper-parameters

In Section 3.2, we have shown that under different noise levels there is an optimal choice for *Top_K* as shown in the plot of metric F1-score. In all experiments above, *Top_K* is either manually set or looped over a range to investigate its effect on the indexing performance. However, for real experimental data sets, *Top_K* should be determined automatically based on the pattern quality to avoid repeated indexing of a single scan. The confidence index (CI) output by the DI step is the dot product between dictionary and experimental pattern, and it is a very good indicator of pattern quality. Thus, with the feedback of DI, an adaptive selection of *Top_K* can be realized. A schematic of this process is shown in Fig. 9.

5. Conclusions

In this paper, a hybrid indexing framework EBSDI-CNN has been proposed for determining crystallographic orientation. It consists of a convolutional neural network followed by dictionary indexing. The computational acceleration of this approach relies on the CNN prediction of probable sub-dictionaries from the entire orientation space. Through a series of experiments on simulated and experimental data, it has been shown that the hybrid approach reduces the computational time by as much as 70% without sacrificing accuracy or robustness against noise. Clearly, this new hybrid framework can greatly expand the application scenarios of the traditional method such as DI to those requiring higher accuracy and indexing speed. As future research become more data-driven, the hybrid strategy of combining CNN and traditional methods will assist researchers to develop faster and more reliable machine learning approaches and explore the latent space of information with ease.

Future studies to improve the current study will focus on realizing transfer learning and few-shot training to further decrease the model training cost, the optimization of the loss function to better quantify the distance between sub-dictionaries, and an algorithm for the adaptive selection of hyper-parameters.

Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgments

General

The authors would like to acknowledge Dr. Maxwell Lee, Dr. Elena Pascal, and Dr. Will Lenthe for many useful discussions and contributions to this work. Dr. Stuart Wright is acknowledged for providing the experimental data set.

Funding

Financial support from a DoD Vannevar-Bush Faculty Fellowship (# N00014-16-1-2821), as well as the computational facilities of the Materials Characterization Facility at CMU under grant # MCF-677785. Use was made of computational facilities purchased with funds from the National Science Foundation (OAC-1925717) and administered by the Center for Scientific Computing (CSC). The CSC is supported by the

California NanoSystems Institute and the Materials Research Science and Engineering Center (MRSEC; NSF DMR 1720256) at UC Santa Barbara.

References

- [1] J.F. Nye, et al., *Physical Properties of Crystals: Their Representation by Tensors and Matrices*, Oxford University Press, 1985.
- [2] B. Larson, W. Yang, G. Ice, J. Budai, J. Tischler, Three-dimensional x-ray structural microscopy with submicrometre resolution, *Nature* 415 (2002) 887–890.
- [3] J.V. Bernier, R.M. Suter, A.D. Rollett, J.D. Almer, High-energy x-ray diffraction microscopy in materials science, *Annu. Rev. Mater. Sci.* 50 (2020) 395–436.
- [4] H. Simons, A. King, W. Ludwig, C. Detlefs, W. Pantleon, S. Schmidt, F. Stöhr, I. Snigireva, A. Snigirev, H.F. Poulsen, Dark-field x-ray microscopy for multiscale structural characterization, *Nat. Commun.* 6 (2015) 1–6.
- [5] A.J. Schwartz, M. Kumar, B.L. Adams, D.P. Field, *Electron Backscatter Diffraction in Materials Science Vol. 2*, Springer, 2009.
- [6] D. Chen, J.-C. Kuo, W.-T. Wu, Effect of microscopic parameters on ebsd spatial resolution, *Ultramicroscopy* 111 (2011) 1488–1494.
- [7] F. Humphreys, Review grain and subgrain characterisation by electron backscatter diffraction, *J. Mater. Sci.* 36 (2001) 3833–3854.
- [8] M. Alam, M. Blackman, D.W. Pashley, *High-Angle Kikuchi Patterns Vol. 221*, The Royal Society London, 1954, pp. 224–242.
- [9] J. Venables, C. Harland, Electron back-scattering patterns—a new technique for obtaining crystallographic information in the scanning electron microscope, *Philos. Mag.* 27 (1973) 1193–1200.
- [10] D. Dingley, On-line determination of crystal orientation and texture determination in an sem, *Proc. R. Microsc. Soc.* 19 (1984) 74–75.
- [11] S.I. Wright, B.L. Adams, Automatic analysis of electron backscatter diffraction patterns, *Metall. Trans. A* 23 (1992) 759–767.
- [12] N.K. Lassen, D.J. Jensen, K. Conradsen, Image processing procedures for analysis of electron back scattering patterns, *Scanning Microsc.* 6 (1992) 115–121.
- [13] P.V. Hough, Method and Means for Recognizing Complex Patterns, 1962. US Patent 3,069,654.
- [14] F. Ram, S. Zaefferer, T. Jäpel, D. Raabe, Error analysis of the crystal orientations and disorientations obtained by the classical electron backscatter diffraction technique, *J. Appl. Crystallogr.* 48 (2015) 797–813.
- [15] F. Ram, S. Wright, S. Singh, M. De Graef, Error analysis of the crystal orientations obtained by the dictionary approach to EBSD indexing, *Ultramicroscopy* 181 (2017) 17–26.
- [16] G. Nolze, Image distortions in sem and their influences on ebsd measurements, *Ultramicroscopy* 107 (2007) 172–183.
- [17] W. Lenthe, S. Singh, M. De Graef, Prediction of potential pseudo-symmetry issues in the indexing of electron backscatter diffraction patterns, *J. Appl. Crystallogr.* 52 (2019).
- [18] E.L. Pang, P.M. Larsen, C.A. Schuh, Resolving pseudosymmetry in tetragonal zro2 using electron backscatter diffraction with a modified dictionary indexing approach, *J. Appl. Crystallogr.* 53 (2020).
- [19] Y.H. Chen, S.U. Park, D. Wei, G. Newstadt, M.A. Jackson, J.P. Simmons, M. De Graef, A.O. Hero, A dictionary approach to electron backscatter diffraction indexing, *Microsc. Microanal.* 21 (2015) 739–752.
- [20] W.C. Lenthe, S. Singh, M.D. Graef, A spherical harmonic transform approach to the indexing of electron back-scattered diffraction patterns, *Ultramicroscopy* 207 (2019) 112841.
- [21] R. Hielscher, F. Bartel, T.B. Britton, Gazing at crystal balls – electron backscatter diffraction indexing and cross correlation on a sphere, *Microsc. Microanal.* 25 (2019) 1954–1955.
- [22] P.G. Callahan, M. De Graef, Dynamical electron backscatter diffraction patterns. part i: pattern simulations, *Microsc. Microanal.* 19 (2013) 1255.
- [23] S. Singh, Y. Guo, B. Winiarski, T.L. Burnett, P.J. Withers, M. De Graef, High resolution low kv ebsd of heavily deformed and nanocrystalline aluminium by dictionary-based indexing, *Sci. Rep.* 8 (2018) 1–8.
- [24] T. Tanaka, N. Maruyama, N. Nakamura, A.J. Wilkinson, Tetragonality of fe-c martensite—a pattern matching electron backscatter diffraction analysis compared to x-ray diffraction, *Acta Mater.* 195 (2020) 728–738.
- [25] A. Winkelman, G. Nolze, G. Cios, T. Tokarski, Mapping of local lattice parameter ratios by projective kikuchi pattern matching, *Phys. Rev. Mater.* 2 (2018) 123803.
- [26] C. Kurniawan, C. Zhu, M. DeGraef, Deformation state extraction from electron backscatter diffraction patterns via simulation-based pattern-matching, *Scr. Mater.* 190 (2021) 147–152.
- [27] D. Roşca, A. Morawiec, M. De Graef, A new method of constructing a grid in the space of 3D rotations and its applications to texture analysis, *Model. Simul. Mater. Sci. Eng.* 22 (2014) 75013.
- [28] S.R. Spurgeon, C. Ophus, L. Jones, A. Petford-Long, S.V. Kalinin, M.J. Olszta, R. E. Dunin-Borkowski, N. Salmon, K. Hattar, W.-C.D. Yang, et al., Towards data-driven next-generation transmission electron microscopy, *Nat. Mater.* (2020) 1–6.
- [29] E.A. Holm, R. Cohn, N. Gao, A.R. Kitahara, T.P. Matson, B. Lei, S.R. Yaras, Overview: computer vision and machine learning for microstructural characterization and analysis, *Metall. Mater. Trans. A* (2020) 1–15.
- [30] S. Ren, K. He, R. Girshick, J. Sun, Faster R-CNN: towards real-time object detection with region proposal networks, *IEEE Trans. Pattern Anal. Mach. Intell.* 39 (2017) 1137–1149.

- [31] B.L. DeCost, B. Lei, T. Francis, E.A. Holm, High throughput quantitative metallography for complex microstructures using deep learning: a case study in ultrahigh carbon steel, *Microsc. Microanal.* 25 (2019) 21–29.
- [32] R.A. Schwarzer, J. Sukkau, Automated evaluation of kikuchi patterns by means of radon and fast fourier transformation, and verification by an artificial neural network, *Adv. Eng. Mater.* 5 (2003) 601–606.
- [33] K. Kaufmann, C. Zhu, A.S. Rosengarten, D. Maryanovsky, H. Wang, K.S. Vecchio, K. S. Vecchio, Phase mapping in EBSD using convolutional neural networks, *Microsc. Microanal.* 26 (2020) 458–468.
- [34] J. Aguiar, M.L. Gong, R. Unocic, T. Tasdizen, B. Miller, Decoding crystallography from high-resolution electron imaging and diffraction datasets with deep learning, *Sci. Adv.* 5 (2019) eaaw1949.
- [35] K. Kaufmann, C. Zhu, A.S. Rosengarten, D. Maryanovsky, T.J. Harrington, E. Marin, K.S. Vecchio, Crystal symmetry determination in electron diffraction using machine learning, *Science* 367 (2020) 564–568.
- [36] K. Kaufmann, C. Zhu, A.S. Rosengarten, K.S. Vecchio, K.S. Vecchio, Deep neural network enabled space group identification in EBSD, *Microsc. Microanal.* 26 (2020) 447–457.
- [37] Y.-F. Shen, R. Pokharel, T.J. Nizolek, A. Kumar, T. Lookman, Convolutional neural network-based method for real-time orientation indexing of measured electron backscatter diffraction patterns, *Acta Mater.* 170 (2019) 118–131.
- [38] D. Jha, S. Singh, R. Al-Bahrani, W.-k. Liao, A. Choudhary, M. De Graef, A. Agrawal, Extracting grain orientations from ebsd patterns of polycrystalline materials using convolutional neural networks, *Microsc. Microanal.* 24 (2018) 497–502.
- [39] Z. Ding, E. Pascal, M. De Graef, Indexing of electron back-scatter diffraction patterns using a convolutional neural network, *Acta Mater.* 199 (2020) 370–382.
- [40] C. Zhu, K. Kaufmann, K. Vecchio, Automated reconstruction of spherical kikuchi maps, *Microsc. Microanal.* 25 (2019) 912–923.
- [41] X. Tao, A. Eades, Errors, artifacts, and improvements in EBSD processing and mapping, *Microsc. Microanal.* 11 (2005) 79–87.
- [42] F. Chollet, Xception: deep learning with depthwise separable convolutions, in: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017, pp. 1251–1258.
- [43] S. Singh, M. De Graef, Orientation sampling for dictionary-based diffraction pattern indexing methods, *Model. Simul. Mater. Sci. Eng.* 24 (2016) 85013.
- [44] I. Goodfellow, Y. Bengio, A. Courville, *Deep Learning*, MIT Press, 2016. <http://www.deeplearningbook.org>.
- [45] S.I. Wright, M.M. Nowell, S.P. Lindeman, P.P. Camus, M. De Graef, M.A. Jackson, Introduction and comparison of new EBSD post-processing methodologies, *Ultramicroscopy* 159 (2015) 81–94.
- [46] D.M. Powers, Evaluation: from precision, recall and f-measure to roc, informedness, markedness and correlation, *arXiv preprint* (2020) arXiv: 2010.16061.
- [47] F. Sung, Y. Yang, L. Zhang, T. Xiang, P.H.S. Torr, T.M. Hospedales, Learning to compare: relation network for few-shot learning, in: *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2018, pp. 1199–1208.
- [48] M. Chapman, S. Singh, M. De Graef, Can EBSD patterns be used for determination of grain boundary inclination? *Microsc. Microanal.* 21 (2015) 2039–2040.
- [49] V. Randle, N. Hansen, D.J. Jensen, The deformation behaviour of grain boundary regions in polycrystalline aluminium, *Philos. Mag. A* 73 (1996) 265–282.
- [50] A.J. Wilkinson, G. Meaden, D.J. Dingley, High resolution mapping of strains and rotations using electron backscatter diffraction, *Mater. Sci. Technol.* 22 (2006) 1271–1278.
- [51] M. De Graef, A dictionary indexing approach for EBSD, *IOP Conf. Ser.: Mater. Sci. Eng.* 891 (2020), 012009.
- [52] M. De Graef, M. Jackson, J. Kleingers, C. Zhu, J. Tessmer, W. Lenthe, S. Singh, M. Atkinson, S. Wright, H. Ånes, EMsoft-org/EMsoft: EMsoft Release 5.0.0, 2019.
- [53] J. Nickolls, W.J. Dally, The GPU computing era, *IEEE Micro.* 30 (2010) 56–69.
- [54] S.W. Keckler, W.J. Dally, B. Khailany, M. Garland, D. Glasco, GPUs and the future of parallel computing, *IEEE Micro.* 31 (2011) 7–17.
- [55] P.M. Larsen, S. Schmidt, Improved orientation sampling for indexing diffraction patterns of polycrystalline materials, *J. Appl. Crystallogr.* 50 (2017) 1571–1582.
- [56] R. Quey, A. Villani, C. Maurice, Nearly uniform sampling of crystal orientations, *J. Appl. Crystallogr.* 51 (2018) 1162–1173.
- [57] K. Weiss, T.M. Khoshgoftaar, D.D. Wang, A survey of transfer learning, *J. Big Data* 3 (2016) 1–40.
- [58] Y. Wang, Q. Yao, J.T. Kwok, L.M. Ni, Generalizing from a few examples: a survey on few-shot learning, *ACM Computing Surveys (CSUR)* 53 (2020) 1–34.
- [59] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, L. Fei-Fei, Imagenet: a large-scale hierarchical image database, in: *IEEE Conference on Computer Vision and Pattern Recognition*, 2009, pp. 248–255.
- [60] A. Krizhevsky, G. Hinton, et al., *Learning Multiple Layers of Features From Tiny Images*, 2009.