ARTICLE OPEN

Data augmentation in microscopic images for material data mining

Boyuan Ma^{1,2,3}, Xiaoyan Wei^{1,2,3}, Chuni Liu^{1,2,3}, Xiaojuan Ban^{1,2,3}✉, Haiyou Huang^{1,3,4}✉, Hao Wang^{1,5}, Weihua Xue^{5,6}, Stephen Wu⁷✉, Mingfei Gao^{1,2,3}, Qing Shen⁸, Michele Mukeshimana⁹, Adnan Omer Abuassba¹⁰, Haokai Shen^{11,12} and Yanjing Su^{1,4}

Recent progress in material data mining has been driven by high-capacity models trained on large datasets. However, collecting experimental data (real data) has been extremely costly owing to the amount of human effort and expertise required. Here, we develop a novel transfer learning strategy to address problems of small or insufficient data. This strategy realizes the fusion of real and simulated data and the augmentation of training data in a data mining procedure. For a specific task of grain instance image segmentation, this strategy aims to generate synthetic data by fusing the images obtained from simulating the physical mechanism of grain formation and the “image style” information in real images. The results show that the model trained with the acquired synthetic data and only 35% of the real data can already achieve competitive segmentation performance of a model trained on all of the real data. Because the time required to perform grain simulation and to generate synthetic data are almost negligible as compared to the effort for obtaining real data, our proposed strategy is able to exploit the strong prediction power of deep learning without significantly increasing the experimental burden of training data preparation.

npj Computational Materials (2020)6:125; <https://doi.org/10.1038/s41524-020-00392-6>

INTRODUCTION

There has been considerable interest over the last few years in accelerating the process of materials design and discovery¹. In the past decade, a new interdisciplinary research field called materials informatics, which combines data and information science with materials science, has led to an increasing number of successful materials discoveries^{2–7}. Machine learning methods have played a key role in many of these studies. In general, under the premise of ensuring data quality, the larger the training dataset is, the more accurate the trained machine learning model becomes. This is especially the case for deep neural networks, which have been shown to have exceptional predictive performance when trained with a large amount of data. As a result, some accelerated methods for data accumulation, such as high-throughput computation and high-throughput experiments, have been developed to build large databases. However, in many materials studies, especially for new materials, we still face the dilemma of lacking high-quality data to construct reliable machine learning models⁸. The major obstacle comes from the typically time-consuming or technically difficult process of collecting experimental data (real data). Although computational data may have a lower cost than experimental data, there is still a significant gap between the two types of data for many applications in materials science.

Material microstructure data are an important type of material data to build the intrinsic relationship between composition, structure, process, and properties that is fundamental to material

design. Therefore, the quantitative analysis of microstructures is essential for the control of the material properties and performances of metals or alloys in various industrial applications^{9,10}. Microscopic images are often used to understand the important structures of a material related to certain properties of interest. One of the key steps in the design process is microscopic image processing using computational algorithms and tools to efficiently extract useful information from the images^{11,12}. For example, image segmentation¹³, which is a task that generates pixel-wise labels of a raw image, is commonly used to extract significant information from microscopic images in the field of material structure characterization¹⁴. For example, in polycrystalline iron (Fig. 1), which is a fundamental and popular subject in practical materials research, the objective of image processing algorithms is to detect and depart each grain from raw microscopic images to obtain accurate descriptions of the microstructure, such as geometric and topological characteristics.

Recent progress in segmentation of material microscopic image^{15–17} has been driven by high-capacity models trained on large datasets. Unfortunately, the generalization performance of these models has been hindered by the lack of a large training data due to the time-consuming labeling of material microscopic images. Generating data of images with pixel-wise semantic labels is known to be very challenging due to the amount of human effort and expertise required to trace the object boundaries accurately¹⁸. Therefore, high quality large training datasets are often difficult to obtain for a new task of material design.

¹Beijing Advanced Innovation Center for Materials Genome Engineering, University of Science and Technology Beijing, Beijing 100083, China. ²School of Computer and Communication Engineering, University of Science and Technology Beijing, Beijing 100083, China. ³Beijing Key Laboratory of Knowledge Engineering for Materials Science, Beijing 100083, China. ⁴Institute for Advanced Materials and Technology, University of Science and Technology Beijing, Beijing 100083, China. ⁵School of Materials Science and Engineering, University of Science and Technology Beijing, Beijing 100083, China. ⁶School of Materials Science and Technology, Liaoning Technical University, Liaoning 114051, China. ⁷The Institute of Statistical Mathematics, Research Organization of Information and Systems, Tachikawa, Tokyo 190-8562, Japan. ⁸National Intellectual Property Administration, Beijing 100088, China. ⁹Faculty of Engineering Sciences, University of Burundi, Bujumbura, Burundi. ¹⁰Faculty of Engineering and Technology, Palestine Technical University – Kadoorie, Tulkarem, Palestine. ¹¹College of Information Science and Engineering, China University of Petroleum, Beijing, China. ¹²Key Lab of Petroleum Data Mining, China University of Petroleum, Beijing, China. ✉email: banxj@ustb.edu.cn; huanghy@mater.ustb.edu.cn; stewu@ism.ac.jp

Fig. 1 Microscopic images of polycrystalline iron. **a** Real experimental image obtained by optical microscopy. **b** Results of manually annotated label of **(a)** conducted by human experts, which was used as the ground truth in our investigation. **c** A 3D image of grain simulation. **d** A slice of 2D image from **(c)**. **e** The simulated label of **(d)**.

In this work, we develop a novel transfer learning strategy to address problems of small or insufficient data in material data mining. This strategy realizes the fusion of real data and simulated/calculated data based on transfer learning theory and the augmentation of training data in a data mining procedure so that a classification or regression model with better performance can be established based on only a small set of real data. We explore the use of a 3D simulated model to create large-scale pixel-accurate data to train image segmentation models. For example, as shown in Fig. 1c, a Monte Carlo Potts model can represent the polycrystalline microstructure of materials because it possesses geometrical and topological microstructure characteristics of grains similar to those of the real grain structure^{19,20}. It is easy to obtain large data with pixel-level labels from a simulated 3D model using computational methods (Fig. 1d, e). However, the acquired contents in the simulated data are unrealistic, due to some theoretical approximations and simplifications in the modeling process, which causes a challenge to simply apply simulated data to a real microscopic image processing system. To fill in the missing information in the simulated data, we use an image-to-image conversion technique to transfer simulated images into synthetic images that incorporates features from real images. This image-to-image conversion can be simply described as an image style transfer model that takes in a simulated image as input and outputs a realistic image^{21,22}. We exploit the transformation power of image style transfer models that input and output images shared the same underlying global structure even if the appearances are very different. Specifically, we pick generative adversarial networks (GANs) as the underlying generative model based on prior studies that demonstrated its outstanding performance on the task of image style transfer^{23–27}.

In general, our presented strategy can create synthetic data by fusing the images obtained from simulating the physical mechanism of grain formation and the “image style” information in real images, as outlined in Fig. 2. The acquired synthetic images are more realistic than the simulated images as they capture features in real images through the transfer learning process. Based on quantitative analyses using the polycrystalline data, we

Fig. 2 Flowchart of the proposed data augmentation strategy.

The presented strategy can create synthetic data by fusing the images obtained from simulating the physical mechanism of grain formation and the “image style” information in real images. Then the acquired synthetic data can be used as data augmentation for the training procedure of the machine learning model.

show that the acquired synthetic data can be used as a partial replacement to the real data when training a segmentation model. In particular, we demonstrate that the model trained with synthetic data and only 35% of the real data can achieve

competitive performance to the model trained on all of the real data. Because the time required to perform grain simulation and to generate synthetic images are almost negligible as compared with the effort for obtaining real data, our proposed strategy is able to exploit the strong prediction power of deep learning without significantly increasing the experimental burden of training data preparation. Besides, we believe that this strategy can be easily applied to other data mining tasks, structured data analysis for example, or even those outside of materials science.

RESULTS

Datasets

We define a data point as a pair of an image and its corresponding label. Real dataset denotes real images taken from experiment with manually annotated labels; simulated dataset refers to Monte Carlo Potts simulated images with labels directly extracted from the simulations; synthetic dataset denotes synthetic images generated from our image style transfer model with corresponding simulated labels. Segmentation models trained with different data sets are used to provide predicted labels for a separated test images obtain from experiment. Throughout the paper, we consider the manually annotated labels as ground true (or true labels) because these are the best references we can get and we have confirmed the consistency of results between different experts.

Real dataset. The real dataset contains a total of 136 serial section optical images of polycrystalline iron with a resolution of 2800×1600 pixels, of which the ground truth has 2 semantic classes (grain and grain boundary) and is manually labeled by material scientists. The dataset is randomly split into 100 training and 36 test data. The original images were pre-processed into small images (patches) with a size of 400×400 pixels in order to reduce the computational burden. Finally, the training set consists of 2800 patches with resolution of 400×400 pixels, cropped from the original 100 training images, while the test set consists of 1008 patches cropped from the 36 test images. The test data is separated at the beginning of our experiment and is not used for any of the model training, including the image style transfer models and segmentation models.

Simulated dataset. We established a large 3D simulated model of the polycrystalline materials using the Monte Carlo Potts model. Then, 2D images were obtained by slicing a simulated 3D image in the normal direction (Fig. 1d). Finally, we retained only the boundary pixels of each grain to obtain the labels for the simulated 2D images (Fig. 1e). The final simulated dataset contains a total of 28,800 images, which was one order of magnitude greater than the real dataset. During the simulation process, we ensured that the geometric and topological information of the simulated data was statistically consistent with that of the real image. However, the simulated data contains only grain boundary information with no defects or noises. For the real images, the range of pixel values of the grain boundary obtained by optical microscopy was $[0, 255]$, and the specific pixel value was affected by grain appearance, the light intensity of the microscope and noise introduced during the sample preparation. For the simulated 2D images, the range of pixel values was $[0, N]$, with N denoting the total number of grains in the 3D simulated model, which was controlled by the grain-growth simulation model. The pixel values are basically the identification numbers for the grains. Simulated data cannot be directly used in machine learning-based algorithms because there are differences in the nature of real and simulated data.

Synthetic dataset. We trained our image style transfer model using real dataset. Then, we applied our model to transform all the labels images of the simulated 2D images (simulated label) into

synthetic images. As shown in Fig. 3, there are four columns of images: from left to right are real image, simulated image, simulated label and synthetic image. The synthetic image has both label information and an “image style” similar to those of the real image. It can be used as data augmentation for the real data in data mining or machine learning tasks.

We compare the time consumption of the two image production methods, i.e., experiment and synthetization, in Table 1. Due to the complex experimental procedures, including sample preparation, polishing, etching, and photographing, the real image takes the longest time, ~ 1200 s per image. It should be noted that we do not consider the increase in time cost caused by failed experiments. In reality, the experimental process is likely to require even longer time.

The preparation of synthetic dataset includes three steps: the design and construction of simulation model, the training of style transfer model, and the generation of synthetic image. By virtue of a high-speed computer system, the construction of simulation model cost only 1% of the experimental time, ~ 12 s per simulated data. And the training of style transfer model cost ~ 23 h, which translated to ~ 3 s per synthetic image. Note that the training time of style transfer model depends on the volume of training data. In practice, a smaller training set could be enough to obtain a reasonably good model, meaning that a shorter time would be needed. Finally, when we have a trained style transfer model, the generation of one synthetic image takes ~ 0.1 s during the inference of the generative adversarial network. As a result, the time cost of setting up a synthetic dataset is approximated 1% of the time cost of getting experimental data.

Evaluation models and metrics

Model setting. There are two deep learning-based models in our experiment: image style transfer model and image segmentation model.

For image style transfer model, we used pix2pix²³ to transform simulated label to synthetic image and the detail structure can be found in latter section. During the training stage, we set the batch size equal to 8 and optimized the model by Adam with an initial learning rate of 2×10^{-4} in 200 epochs.

For image segmentation model, U-net²⁸ is the most commonly used supervised learning method in the field of materials and medical image processing²⁹. Therefore, we used U-net as our baseline to compare with data augmentation algorithms. U-net is an encoder-decoder network; the input goes through a series of convolution-pooling-normalization group layers until it reaches the bottleneck layer, where the underlying information is shared with the output. U-net joins the layer-skip connection to transfer the features extracted from the down-sampling layers directly to the upper sampling layers, thus making the pixel location of the network more accurate. During the training stage, we jointly trained the model on real and synthetic dataset using batch gradient descent with mini-batches of 8 images, including 4 real and 4 synthetic images, which is the same process used in the work¹⁸. It took 28,000 iterations to converge with a learning rate of 1×10^{-4} . Although the training pattern is important for network training, this paper does not discuss this issue because it is beyond the scope of our topic. All models were trained with the same pattern to ensure fairness.

Metrics. Grain segmentation is an instance segmentation (or dense segmentation) task. A successful algorithm has to detect and depart each grain in an image. For example, there are ~ 57 grains in each 400×400 real image. Based on the segmentation results, researchers can extract size and shape distribution of grains from the image to build the relationship between microstructure and macroscopic material performance. In practice, different types of noises are introduced to a sample during the

Fig. 3 The demonstration of different datasets. From left to right are the real image, simulated image, simulated label, and synthetic image.

Table 1. Manufacture time of two procedures.				
Procedures	Sample real experimental image	Synthesize		
		Construct simulated model	Train style transfer model	Generate synthetic image data
Operation time (s) per image with a size of 400 × 400 pixels	~1200	~12	~3	~0.1

preparation step, which seriously influences the grain segmentation algorithms. We should choose an effective metrics to evaluate the performance of algorithms in this task. Discussion of different image segmentation metrics is included in Supplementary Note 1 with Supplementary Fig. 1. We decided to use two effective metrics to evaluate our algorithm: mean average precision (MAP)^{30,31} and adjusted rand index (ARI)^{32–34}.

MAP is a classical measure in image segmentation and object detection tasks. In this paper, we evaluate it at different intersection over union (IoU) thresholds. The IoU of a proposed set of object pixels and a set of true object pixels is calculated as:

$$\text{IoU}(A, B) = \frac{A \cap B}{A \cup B}$$

(1)

The metric sweeps over a range of IoU thresholds at each point, calculating an average precision value. The threshold values range from 0.5 to 0.95 with a step size of 0.05: (0.5, 0.55, 0.6, 0.65, 0.7,

0.75, 0.8, 0.85, 0.9, 0.95). In other words, at a threshold of 0.5, a predicted object is considered a “hit” if its IoU with a ground truth object is >0.5. Generally, it can be considered that the segment is correct when its IoU is >0.5. The other higher value is aimed at ensuring the correct results.

At each threshold value t , a precision value is calculated based on the number of true positives (TP), false negatives (FN), and false positives (FP) resulting from comparing the predicted object to all ground truth objects. The average precision of a single image is then calculated as the mean of the above precision values at each IoU threshold.

$$\text{Average precision} = \frac{1}{|\text{thresholds}|} \sum_t \frac{\text{TP}(t)}{\text{TP}(t) + \text{FP}(t) + \text{FN}(t)}$$

(2)

Finally, the MAP score returned by the metric is the mean taken over the individual average precisions of each image in the test set.

Table 2. Contingency table.

	Y_1	Y_2	...	Y_s	Sums
X_1	n_{11}	n_{12}	...	n_{1s}	a_1
X_2	n_{21}	n_{22}	...	n_{2s}	a_2
...	...	...	...	...	...
X_r	n_{r1}	n_{r2}	...	n_{rs}	a_r
Sums	b_1	b_2	...	b_s	

ARI is the corrected-for-chance version of the rand index (RI), which is a measure of the similarity between two data clustering methods^{32–34}. From a mathematical standpoint, ARI or RI is related to accuracy. In addition, image segmentation can be considered a clustering task that splits all pixels in an image into n partitions or segments.

Given a set S of n elements (pixels) and two groupings or partitions of these elements, namely, $X = \{X_1, X_2, \dots, X_r\}$ (a partition of S into r subsets) and $Y = \{Y_1, Y_2, \dots, Y_s\}$ (a partition of S into s subsets), the overlap between X and Y can be summarized in a contingency table $[n_{ij}]$ (see Table 2), where each entry n_{ij} denotes the number of objects in common between X_i and Y_j : $n_{ij} = |X_i \cap Y_j|$. For the image segmentation task, X and Y can be treated as ground truth and predicted result, respectively.

The ARI is defined as follows:

$$\widehat{\text{ARI}} = \frac{\sum_{ij} \binom{n_{ij}}{2} - \left[\sum_i \binom{a_i}{2} \sum_j \binom{b_j}{2} \right] / \binom{n}{2}}{\frac{1}{2} \left[\sum_i \binom{a_i}{2} + \sum_j \binom{b_j}{2} \right] - \left[\sum_i \binom{a_i}{2} \sum_j \binom{b_j}{2} \right] / \binom{n}{2}} \quad (3)$$

where n_{ij} , a_i , b_j are values from the contingency table and $\binom{n}{2}$ is calculated as $n(n-1)/2$.

The MAP is a strict and effective metric because it is the mean score of each threshold, so the score is always smaller than the ARI.

For all models, the higher the metrics, the better the model is. For fair comparison, all the models were evaluated on the same real test set initially separated out from the training process.

Our implementation of this algorithm was derived from the publicly available Python³⁵, Pytorch framework³⁶, and OpenCV toolbox³⁷. The image style transfer model and U-net's training and testing were performed on Nvidia DGX Station using 4 NVIDIA Tesla V100 GPU with 128 GB memory.

Image segmentation by the proposed augmentation method

First, we compared the performance of three conventional image processing algorithms; a threshold-based method called OTSU³⁸, a boundary-based method called Canny³⁹ and a morphology-based method called Watershed⁴⁰. Second, we chose K-means⁴¹ as a representative for unsupervised learning method. Third, we chose U-net as a representative for deep learning method (Real_{100%} in Table 3). As shown in Table 3, due to the high capability of feature extraction, the U-net achieves the best performance compared with unsupervised learning method and traditional image processing algorithms. And some visualization results can be found in Supplementary Note 2 with Supplementary Fig. 2.

Finally, we explored the use of a simulated dataset as data augmentation to replace the real dataset for training a neural network. As shown in Table 3, using the same real test set initially separated out from the training, we compared the performance of the model trained on the whole training set of real data, named the Real_{100%} dataset, and the performance on the whole simulated dataset, named the Simulated_{100%} dataset. The subscript denotes the percentage of the specific dataset used in the training set. We found that simply using the Simulated_{100%} dataset achieved poor

Table 3. The performance of different methods in image segmentation.

Methods	MAP	ARI
Traditional image processing		
OTSU	0.198	0.204
Canny	0.464	0.276
Watershed	0.263	0.646
Unsupervised learning		
K-Means	0.198	0.204
Supervised learning (U-net)		
Real _{100%}	0.585	0.866
Simulated _{100%}	0.112	0.078
Mixed (Real _{100%} + Simulated _{100%})	0.504	0.752

Bold values indicate the best performance in each metric.

performance when compared with the Real_{100%} dataset. In addition, the performance was degraded when we mixed these two datasets as a training set. We assume that there are two possible reasons. One possibility is that we did not have sufficient simulated data to achieve proper model training. However, additional experiments (Supplementary Table 1) shown in Supplementary Note 3 indicated that models trained with simulated data performed very well on the simulated test data, so we ruled out this possibility. The other reason could be that the simulated data were unrealistic, i.e., containing only grain boundary information without any defects that may appear in real images. This problem can be addressed by the image style transfer model.

We used the image style transfer model to create synthetic image by fusing the pixel-level label of simulated label and the "image style" information of real image. We believe that this processing will make the simulated data more realistic and could be used as data augmentation for real data in material data mining.

We evaluated the performance of the image style transfer model using MAP and ARI. There are two stages in our approach; image style transfer and image segmentation, both of which need to be trained. The main motivation of our study is to reduce the demand of real data for training the segmentation network. Therefore, we started our experiment from using only 5% of the whole real datasets and increased the amount by 5% until we reached the same performance as the Real_{100%} case. First, we used the selected amount of real data to train an image style transfer model. Then, we used the trained image style transfer model to convert all images of simulated label to a set of synthetic images. The simulated labels and synthetic images form the synthetic dataset, denoted as Synthetic_{100%}^{5%} if we used 5% real data. The superscript in Synthetic_{100%}^{5%} denotes the proportion of real dataset used to train the style transfer model and the subscript of Real_{5%} or Synthetic_{100%}^{5%} denotes the proportion of real or synthetic dataset, respectively, used to train the segmentation network.

We show the performance of model trained with synthetic dataset in Table 4. We compared the performance with models trained on real dataset and a mixture of both. We found that the case of using mixed dataset performed better than the case of using only real dataset. If we have only 5% of the real dataset, the performance will increase by ~8% on MAP and 10% on ARI after using the synthetic dataset as data augmentation. This finding suggests that using synthetic dataset as data augmentation will

bring significant performance improvement, especially when there is only a small amount of real data.

As shown in Table 4, with the increase in the amount of real data, the performance of the models improved for both cases (using only the real data or using the mixed data for training). When we used 35% of the real data, the performance (MAP is 0.586 and ARI is 0.875) of the case of mixed training ($\text{Real}_{35\%} + \text{Synthetic}_{100\%}^{35\%}$) achieved competitive performance to the case of using the whole real data ($\text{Real}_{100\%}$) in both metrics. This finding proves that our method can significantly reduce the amount of real data in image segmentation (around 65% in this case), which means reducing the pressure of obtaining and labeling real images from costly experiment. In order to demonstrate the statistical significance of resulted values, we have conducted extra tests to estimate the mean and standard deviation of the

Table 4. Performance of synthetic dataset used as data augmentation in image segmentation.

Dataset	MAP	ARI	Dataset	MAP	ARI
$\text{Real}_{5\%}$	0.481	0.735	Mixed($\text{Real}_{5\%} + \text{Synthetic}_{100\%}^{5\%}$)	0.560	0.837
$\text{Real}_{10\%}$	0.506	0.774	Mixed($\text{Real}_{10\%} + \text{Synthetic}_{100\%}^{10\%}$)	0.572	0.851
$\text{Real}_{15\%}$	0.531	0.800	Mixed($\text{Real}_{15\%} + \text{Synthetic}_{100\%}^{15\%}$)	0.574	0.853
$\text{Real}_{20\%}$	0.542	0.830	Mixed($\text{Real}_{20\%} + \text{Synthetic}_{100\%}^{20\%}$)	0.582	0.858
$\text{Real}_{25\%}$	0.534	0.832	Mixed($\text{Real}_{25\%} + \text{Synthetic}_{100\%}^{25\%}$)	0.579	0.870
$\text{Real}_{30\%}$	0.548	0.837	Mixed($\text{Real}_{30\%} + \text{Synthetic}_{100\%}^{30\%}$)	0.586	0.863
$\text{Real}_{35\%}$	0.559	0.836	Mixed($\text{Real}_{35\%} + \text{Synthetic}_{100\%}^{35\%}$)	0.586	0.875

The superscript of $\text{Synthetic}_{Y\%}^{X\%}$ means to use $X\%$ of real dataset to train the style transfer network and the subscript of $\text{Real}_{Y\%}$ and $\text{Synthetic}_{Y\%}^{X\%}$ means to use $Y\%$ of specific dataset (real or synthetic) to train the segmentation network. The same real test set, which is initially separated out from the training process, is used for the evaluation of MAP and ARI for each experiment.

performance of our best model (see Supplementary Note 4 with Supplementary Table 2).

The visualization of image segmentation is shown in Fig. 4. From left to right are real image, manually annotated label, the results of model training with $\text{Real}_{35\%}$ and the results of model training with mixed data ($\text{Real}_{35\%} + \text{Synthetic}_{100\%}^{35\%}$). The orange rectangles and their magnified results (shown in upper left of the figure) denote the areas in which the model trained with small real data cannot close the grain boundary, but the model trained with data augmentation can segment those areas correctly. The experimental results demonstrate that the proposed data augmentation method improves the performance in terms of quantitative assessment and result visualization.

In addition, we compare the performance of our data augmentation strategy with traditional augmentation methods (such as rotation and flipping) and pre-training method (Ternausnet⁴²) in Supplementary Notes 5 and 6 with Supplementary Tables 3, 4, respectively. We confirmed that our proposed augmentation method provides extra improvement to the segmentation task on top of the traditional data augmentation methods. Furthermore, the typical pre-training method (e.g., Ternausnet) did not achieve the same level of improvement as our proposed method did.

DISCUSSION

Modern machine learning methods, such as deep learning, have outstanding prediction performance when the model is trained with a sufficiently large amount of data. However, for most applications in materials science, there is a chronic lack of experimental data for any given task, i.e., the small data dilemma. In the present work, we developed a novel transfer learning strategy to address problems of small or insufficient data in material data mining. This strategy realizes the fusion of experimental data (real data) and simulated/calculated data based on transfer learning theory and the augmentation of training data in a data mining procedure, so that classification or regression models with better performance can be established based on only a small set of real data. Then, we applied this strategy to the task of grain instance segmentation. First, we transferred the “image

Fig. 4 The results of image segmentation with different training sets. From left to right are real image, manually annotated label (ground truth), the result of model training with $\text{Real}_{35\%}$, and the result of model training with Mixed($\text{Real}_{35\%} + \text{Synthetic}_{100\%}^{35\%}$).

style" information of a real image to an image of simulated label to generate a synthetic image. Such a transfer learning technique extracted important features from the real image to modify the images from simulation, making them more relevant to the segmentation task of real images⁴³. Second, by supplementing a machine learning model with synthetic dataset (synthetic image and simulated label), the model captured useful features from the synthetic dataset that are transferable to the real dataset and achieved a promising performance.

Computational simulation is a good way to acquire data efficiently. We assumed that the Monte Carlo Potts model could generate simulated data that mimic the pattern of grain growth under ideal conditions. In materials science, simulated data partially capture the actual physical mechanism. However, the Monte Carlo Potts model could not simulate the perturbation of data in practical experiments, mainly coming from the process of sample preparation in the experiment. As shown in Table 3, when there is a large difference between the simulated data and the real data, it is not enough to simply mix the simulated dataset into the real dataset, which may sometimes lead to a negative effect.

Recently, generative adversarial networks (GAN), a deep learning based generative model has been proposed to learn the data distribution and achieve promising performance in image generation. It can learn a loss that tries to classify whether the output image is real or fake while simultaneously training a generative model to minimize this loss⁴⁴. We chose it as the primary architecture behind our image style transfer model. We exploit the generative power of GAN to capture the small perturbation of real data that commonly occurs in actual experiments. The image style transfer model, then, generates synthetic images by mixing in realistic features into the images from computational simulation.

By treating the synthetic dataset as data augmentation for a machine learning model, the model achieves a promising performance when trained on a mix of 35% real dataset and the whole synthetic dataset, which is competitive to the performance of the model trained on the whole real dataset. In addition, as the time required to generate the synthetic dataset is almost negligible when compared with the time required to generate real dataset, our method is able to achieve high-quality segmentation without a high cost of preparing training data. Hence, this strategy can be used to address the problems of small or insufficient data in material data mining.

This paper has demonstrated the viability of the combination of simulated and real experimental data, suggesting that simulated data (after performing image style transfer from the real data) could be useful in data mining or machine learning problems. We believed that the proposed strategy could be easily applied to other data mining tasks, structured data analysis, or even those outside materials science. Our next step will be to investigate a more efficient transfer learning technique for this segmentation task to fully suppress the need to use real experimental dataset for training.

METHODS

Experimental

A commercial hot-rolled iron plate with a purity of >99.9 wt% was used in this work. The plate was forged into a round bar with a diameter of 30 mm and then fully recrystallized by annealing at 1153 K for 3 h. The samples for metallographic characterization were spark cut from the bar. One surface of the samples was mechanically polished for a fixed time, and metallographic photographs were then taken with an optical microscope after etching using a 4 vol% nital solution. The steps of the polish-etching photograph above were repeated to obtain serial section photographs. The average thickness interval between the two sections is ~1.8 μm.

Monte Carlo Potts simulated model

The Monte Carlo Potts model^{19,20} was used to simulate three-dimensional normal grain growth and the processed images were exported as simulated data to train segmentation models. We used a $400 \times 400 \times 400$ cubic lattice with full periodic boundary conditions to represent the continuum microstructure. A positive integer, S_i , termed an index number, was assigned to each site in the lattice sequentially. The index number of a site corresponded to the orientation of the grain that it belonged to. Sites with the same index were considered part of the same grain, and the grain boundary existed only between neighbors with different orientations. The Potts model served as the grain boundary energy function:

$$E = -J \sum_{i=1}^N \sum_{j=1}^{N_N} (\delta_{S_i, S_j} - 1), \delta_{S_i, S_j} = \begin{cases} 1, S_i = S_j \\ 0, S_i \neq S_j \end{cases} \quad (4)$$

where E is the boundary energy, N is the system size (the total sites in the system), J is the positive constant that scales the boundary energy, N_N is the number of nearest neighbors of site i , and δ is the Kronecker function with $\delta_{S_i, S_j} = 1$ if $S_i = S_j$ and 0 otherwise. Here, $N_N = 26$, which means that the interactions between a given site and its 6 first-nearest neighbor sites, 12 second-nearest neighbor sites, and 8 third-nearest neighbor sites were considered in the calculation of grain boundary energy. The simulation of grain growth involved a reorientation attempt for each site. Thus, the net energy change associated with the reorientation trial can be expressed by:

$$\Delta E_i = E_{i2} - E_{i1} \quad (5)$$

where E_{i1} and E_{i2} are the boundary energies of site i before and after the reorientation trial. The possibility of this reorientation is given by:

$$W = \begin{cases} 1, & \Delta E \leq 0 \\ \exp\left(-\frac{\Delta E}{k_B T}\right), & \Delta E > 0 \end{cases} \quad (6)$$

where k_B is the Boltzmann constant and T is the Monte Carlo temperature. The term $k_B T$ defines the thermal fluctuations within the simulated system. Time is scaled by the Monte Carlo step (MCS), which corresponds to as many orientation trials as there are lattice sites.

The probability of the trial reorientation for the target is determined by Eq. (3) with the constant $\left(\frac{k_B T}{J}\right) = 0.5$, which is large enough to reduce lattice pinning but small enough to avoid lattice break-up.

Figure 1c shows the Monte Carlo simulation structure. Figure 5 shows the distribution of grain area and the distribution of the number of grain edge. The pure iron structure and Monte Carlo simulation structure have features that are similar in terms of grain morphology, grain area distribution, and edge number distribution.

Image style transfer

We used the image style transfer model to generate simulated data by extracting the "image style" information from real images. Specifically, the conditional GAN (pix2pix)^{23,45} was adopted to carry out the image-to-image transformation. A GAN consists of two modules: a generator G and a discriminator D . Assume that x is the source image and y is the target image such that (x, y) is a pair of data, for example, two styles of an image. The generator G produces an output image $G(x)$ given the original image x before style transfer as a condition, while the discriminator D distinguishes $G(x)$ from the corresponding y . Both modules are optimized by adversarial training to make the "fake" $G(x)$ closer to the "true" y .

The objective function of a conditional GAN can be expressed as:

$$L_{\text{CGAN}}(G, D) = E_{x,y} [\log D(x, y)] + E_x [\log (1 - D(x, G(x)))] \quad (7)$$

where G tries to minimize this objective and D tries to maximize it.

$$G^* = \arg \min_G \max_D L_{\text{CGAN}}(G, D) \quad (8)$$

A previous study²¹ indicated that the generator has to not only fool the discriminator but also be similar to the ground truth output in an L1 sense. The L1 loss is described as follows:

$$L_1(G) = E_{x,y} [\|y - G(x)\|_1] \quad (9)$$

The final objective is as follows:

$$G^* = \arg \min_G \max_D L_{\text{CGAN}}(G, D) + \lambda L_1(G) \quad (10)$$

We performed image-to-image transfer to convert image of a simulated label to a synthetic image, as shown in Fig. 6. Our generator used the encoding-decoding network U-net²⁸. The network structure bridges the

Fig. 5 The compared distribution of characteristic of experimental and simulated data. **a** The distribution of grain area. **b** The distribution of grain edge.

Fig. 6 A two-stage illustration of the style transfer model. The first row shows the training stage of the style transfer model and the second row shows the inference stage.

input and output through a bottleneck layer, which helps to retain the underlying structural similarity in a lower dimensional space. The discriminator calculates the loss of local patches between the output and ground truth to represent the consistency of local details⁴⁶. In the training stage, the generator is given an image of manually annotated label as input and is trained to produce a synthetic image, similar to the real image. The discriminator, on the other hand, is trained to distinguish the “real ones” versus the generated “synthetic ones”. During the training, the generator acquires the ability to transfer the input labels to synthetic images. Then, we use this generative model to perform inference on the images of simulated label, i.e., converting all images of simulated labels into synthetic images.

DATA AVAILABILITY

The datasets and codes are available from the corresponding author on reasonable request. The third-party image style transfer codes that we used are available online⁴⁵.

Received: 10 December 2019; Accepted: 28 July 2020;
Published online: 18 August 2020

REFERENCES

- Lookman, T., Alexander, F. J. & Rajan, K. *Information Science for Materials Discovery and Design*. (Springer Cham. Press, Switzerland, 2016).
- Butler, K. T., Davies, D. W., Cartwright, H., Isayev, O. & Walsh, A. Machine learning for molecular and materials science. *Nature* **559**, 547–555 (2018).
- Raccuglia, P. et al. Machine-learning-assisted materials discovery using failed experiments. *Nature* **533**, 73–76 (2016).
- Luna, P. D., Wei, J., Bengio, Y., Aspuru-Guzik, A. & Sargent, E. Use machine learning to find energy materials. *Nature* **552**, 23–27 (2017).
- Sanchez-Lengeling, B. & Aspuru-Guzik, A. Inverse molecular design using machine learning: Generative models for matter engineering. *Science* **361**, 360–365 (2018).

6. Wu, S. et al. Machine-learning-assisted discovery of polymers with high thermal conductivity using a molecular design algorithm. *npj Comput. Mater.* **5**, 66 (2019).
7. Wang, C. S., Fu, H. D., Jiang, L., Xue, D. Z. & Xie, J. X. A property-oriented design strategy for high performance copper alloys via machine learning. *npj Comput. Mater.* **5**, 87 (2019).
8. Wu, Y. J., Fang, L. & Xu, Y. B. Predicting interfacial thermal resistance by machine learning. *npj Comput. Mater.* **5**, 56 (2019).
9. Dursun, T. & Soutis, C. Recent developments in advanced aircraft aluminium alloys. *Mater. Des.* **56**, 862–871 (2014).
10. Hu, J., Shi, Y. N., Sauvage, X., Sha, G. & Lu, K. Grain boundary stability governs hardening and softening in extremely fine nanograined metals. *Science* **355**, 1292–1296 (2017).
11. Sonka, M., Hlavac, V. & Boyle, R. *Image Processing, Analysis, and Machine Vision*. 4th edn (Cengage Learning, Press, Boston, 2014).
12. Krizhevsky, A., Sutskever, I. & Hinton, G. E. Imagenet classification with deep convolutional neural networks. *NeurIPS*. 1097–1105 (2012).
13. Long, J., Shelhamer, E. & Darrell, T. Fully convolutional networks for semantic segmentation. *CVPR*. 3431–3440 (2015).
14. Lewis, A. C. & Howe, D. Future directions in 3D materials science: outlook from the first international conference on 3D materials science. *JOM* **66**, 670–673 (2014).
15. Li, W., Field, K. G. & Morgan, D. Automated defect analysis in electron microscopic images. *npj Comput. Mater.* **4**, 35 (2018).
16. Azimi, S. M., Britz, D., Engstler, M., Fritz, M. & Mucklich, F. Advanced steel microstructural classification by deep learning methods. *Sci. Rep.* **8**, 2128 (2018).
17. Ma, B. Y. et al. Deep learning-based image segmentation for al-la alloy microscopic images. *Symmetry* **10**, 107 (2018).
18. Richter, S. R., Vineet, V., Roth, S. & Koltun, V. Playing for data: ground truth from computer games. *ECCV*. 102–118 (2016).
19. Anderson, M. P., Grest, G. S. & Srolovitz, D. J. Computer simulation of normal grain growth in three dimensions. *Philos. Mag. B* **59**, 293–329 (1989).
20. Radhakrishnan, B. & Zacharia, T. Simulation of curvature-driven grain growth by using a modified Monte Carlo algorithm. *Metall. Mater. Trans. A* **26**, 167–180 (1995).
21. Gatys, L. A., Ecker, A. S. & Bethge, M. Image style transfer using convolutional neural networks. *CVPR*. 2414–2423 (2016).
22. Gatys, L. A., Ecker, A. S. & Bethge, M. A neural algorithm of artistic style. Preprint at <https://arxiv.org/abs/1508.06576> (2015).
23. Isola, P., Zhu, J. Y., Zhou, T. & Efros, A. A. Image-to-image translation with conditional adversarial networks. *CVPR*. 1125–1134 (2017).
24. Zhu, J. Y., Park, T., Isola, P. & Efros, A. A. Unpaired image-to-image translation using cycle-consistent adversarial networks. *ICCV*. 2223–2232 (2017).
25. Shrivastava, A. et al. Learning from simulated and unsupervised images through adversarial training. *CVPR*. 2107–2116 (2017).
26. Choi, Y. et al. Stargan: unified generative adversarial networks for multi-domain image-to-image translation. *CVPR*. 8789–8797 (2018).
27. Zhu, J. Y. et al. Toward multimodal image-to-image translation. *NeurIPS*. 465–476 (2017).
28. Ronneberger, O., Fischer, P. & Brox, T. U-net: convolutional networks for biomedical image segmentation. *MICCAI*. 234–241 (2015).
29. Falk, T. et al. U-Net: deep learning for cell counting, detection, and morphometry. *Nat. Methods* **16**, 67–70 (2019).
30. Kaggle data science bowl 2018. <https://www.kaggle.com/c/data-science-bowl-2018/overview/evaluation> (2019).
31. Lin, T. Y. et al. Microsoft COCO: common objects in context. *ECCV*. 740–755 (2014).
32. Rand, W. M. Objective criteria for the evaluation of clustering methods. *J. Am. Stat. Assoc.* **66**, 846–850 (1971).
33. Hubert, L. & Arabie, P. Comparing partitions. *J. Classif.* **2**, 193–218 (1985).
34. Vinh, N. X., Epps, J. & Bailey, J. Information theoretic measures for clusterings comparison: Variants, properties, normalization and correction for chance. *JMLR* **11**, 2837–2854 (2010).
35. Python language reference. <http://www.python.org> (2019).
36. Pytorch. <https://pytorch.org/> (2019).
37. Laganà, R. *OpenCV 3 Computer Vision Application Programming Cookbook*. 3rd edn (Packt Publishing Ltd. Press, Birmingham, 2017).
38. Otsu, N. A threshold selection method from gray-level histograms. *IEEE T. Syst. Man Cybern.* **9**, 62–66 (1979).
39. Canny, J. A computational approach to edge detection. *TPAMI* **6**, 679–698 (1986).
40. Roerdink, J. B. & Meijster, A. The watershed transform: definitions, algorithms and parallelization strategies. *Fund. Inform.* **41**, 187–228 (2000).
41. Jain, A. K. Data clustering: 50 years beyond K-means. *ICPR* **31**, 651–666 (2010).
42. Iglovikov, V. & Shvets, A. Terausnet: U-net with vgg11 encoder pre-trained on imagenet for image segmentation. Preprint at <https://arxiv.org/abs/1801.05746> (2018).
43. Yamada, H. et al. Predicting materials properties with little data using shotgun transfer learning. *ACS Central Sci.* **5**, 1717–1730 (2019).
44. Goodfellow, I. J. et al. Generative adversarial. *Netw. NeurIPS* **3**, 2672–2680 (2014).
45. Pix2Pix in Pytorch. <https://github.com/junyanz/pytorch-CycleGAN-and-pix2pix> (2019).
46. Li, C. & Wand, M. Precomputed real-time texture synthesis with markovian generative adversarial networks. *ECCV*. 702–716 (2016).

ACKNOWLEDGEMENTS

The authors acknowledge financial support from the National Key Research and Development Program of China (No. 2016YFB0700500), the National Science Foundation of China (No. 51574027, No. 61572075, No. 6170203, No. 61873299), the Finance science and technology project of Hainan province (No. ZDYF2019009), and the Fundamental Research Funds for the University of Science and Technology Beijing (No. FRF-BD-19-012A, No. FRF-TP-19-043A2). The computing work is partly supported by USTB MatCom of Beijing Advanced Innovation Center for Materials Genome Engineering. And this manuscript was edited for English language by Springer Nature. This work was supported by Beijing Top Discipline for Artificial Intelligent Science and Engineering, University of Science and Technology Beijing.

AUTHOR CONTRIBUTIONS

B.M. conceived the idea, designed the experiment and wrote the paper; X.W. participated in the paper writing and conducted experiments with C.L.; X.B. organized the research project; H.H. participated in the experimental design and discussion with S.W.; H.W. and W.X. prepared the data for the experiments. M.G. prepared the computational environment of the experiment. Y.S., Q.S., M.M., A.O.A., and H.S. were involved in the analyses of data. Besides, Y.S. provided financial support.

COMPETING INTERESTS

B.M., X.W., X.B., H.H., H.W., and W.X. declare the following competing interests: one patent has been registered (201910243002.6).

ADDITIONAL INFORMATION

Supplementary information is available for this paper at <https://doi.org/10.1038/s41524-020-00392-6>.

Correspondence and requests for materials should be addressed to X.B., H.H. or S.W.

Reprints and permission information is available at <http://www.nature.com/reprints>

Publisher's note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Open Access This article is licensed under a Creative Commons Attribution 4.0 International License, which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons license, and indicate if changes were made. The images or other third party material in this article are included in the article's Creative Commons license, unless indicated otherwise in a credit line to the material. If material is not included in the article's Creative Commons license and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this license, visit <http://creativecommons.org/licenses/by/4.0/>.

© The Author(s) 2020