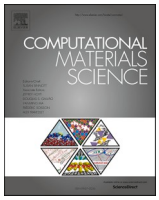
Computational Materials Science 226 (2023) 112250

Contents lists available at ScienceDirect

Computational Materials Science

journal homepage: www.elsevier.com/locate/commatsci



Full Length Article

Data augmentation in material images using the improved HP-VAE-GAN

Yuexing Han [a,b,c], Yuhong Liu [a], Qiaochuan Chen [a,*]

[a] School of Computer Engineering and Science, Shanghai University, Shanghai 200444, China
[b] Key Laboratory of Silicate Cultural Relics Conservation (Shanghai University), Ministry of Education, Shanghai 200444, China
[c] Zhejiang Laboratory, Hangzhou 311100, China

ARTICLE INFO

Keywords:
Data augmentation
Image generation
HP-VAE-GAN
CBAM
Material image classification

ABSTRACT

Since the rapid development of computer vision relies heavily on large-scale labeled data and high-performance computing equipment, therefore, image recognition in small sample datasets faces several challenges, such as difficult to implement model training. In the field of materials research, the cost of collecting image data is relatively high. In order to solve the problem of insufficient image samples in material research, an improved HP-VAE-GAN is proposed to generate material images to achieve data augmentation. HP-VAE-GAN is a single sample generation model that consists of Patch-VAE and Patch-GAN. The improved HP-VAE-GAN introduces the attention mechanism into model. By adding CBAM (Convolutional Block Attention Module) to the encoder of Patch-VAE, the feature extraction and representation capabilities of the network are further improved. Use this model to train a single image, and then generate a certain number of samples to achieve the expansion of the training set. For the classification of ultrahigh carbon steel microstructure images, experiments show that the accuracy of classification model (MobileNet, ResNet50 and VGG16) trained with real images plus generated images is improved obviously. In addition, the effectiveness of the improved HP-VAE-GAN is verified by experiments on texture images similar to material images.

1. Introduction

In the past few years, machine learning has been widely used in various interdisciplinary fields, such as material informatics, and has achieved some excellent results. Material informatics emphasizes the cross-study of the composition, structure, process, and performance of material. With the deepening of study, material data plays an increasingly crucial role in material science, and material images become the driving force for materials research [1]. Generally, machine learning models usually need to be trained on plenty of data to achieve high accuracy, while deep learning models also require a large number of training samples. However, in the specific field of material informatics, there is a small sample dilemma. Due to the limitations of collection cost, privacy protection, confidentiality scope, etc., high-quality images are lacking in material research. Thus, it is difficult to build reliable machine learning models or deep learning models.

While the datasets in some computer vision tasks have few samples, if some annotated data are added to the original dataset, we can use the extended dataset to assist the model learning for the target task. Image data augmentation [2,3] is just such an effective method. This approach can be classified into image-based augmentation and feature-based augmentation. There are many other image augmentation methods besides the traditional methods of color or geometric transformation of images. A particularly popular method is to use GAN (Generative Adversarial Networks) [4] for image generation. GAN is also used in many fields, such as image generation, style transfer, and image super-resolution reconstruction [5]. Image generation using GAN requires a large amount of data to train the model. Since the essence of GAN is to fit the distribution of real data, GAN cannot achieve the expected results for datasets with small samples. The quality of images generated based on poor data distribution is not high. VAE (Variational Autoencoder) [6], another generative model based on a self-coding structure, also has similar problems. Training the VAE model also requires a large amount of training data. Images generated using VAE are usually blurry. In the field of material research, material microscopic images generally have complex textures. Blurred images generated by VAE with low resolution are generally not acceptable because a lot of texture information is lost in these images. While the methods based on feature level such as SMOTE [7], SamplePairing [8], and Mixup [9] all tried to continuum the discrete sample to fit the real sample distribution. However, the added

* Corresponding author at: School of Computer Engineering and Science, Shanghai University, 99 Shangda Road, Shanghai 200444, China.
E-mail addresses: han_yx@i.shu.edu.cn (Y. Han), lyhstu@shu.edu.cn (Y. Liu), qcchen@shu.edu.cn (Q. Chen).

https://doi.org/10.1016/j.commatsci.2023.112250
Received 10 March 2023; Received in revised form 8 April 2023; Accepted 1 May 2023
Available online 10 May 2023
0927-0256/

samples from these methods are still located in the region enclosed by the original sample points in the feature space. It might be better to extend the data beyond this region. For the recognition of small sample images in material research, it is difficult to obtain satisfactory results by augmenting the image in the feature space.

In order to accurately identify material images when samples are scarce, we propose to augment the training set of material images with images generated using the modified HP-VAE-GAN. Unlike multiple sample generative models, the training data of HP-VAE-GAN consists of only a single image, so it does not require a large number of training samples. The improved HP-VAE-GAN adds CBAM (Convolutional Block Attention Module) [11] to the encoder, which improves the quality of the generated images after model training. The feature maps output by CBAM will get attention weights in both channel and space dimension, which makes the connections between each feature more closely in both channel and space. The feature representation capability of the network is enhanced and the resulting feature maps are finer, which also makes the generated images of higher quality.

The main contributions of this work are summarized as follows:

i. The HP-VAE-GAN model is improved. By adding CBAM to the model, the network is able to not only learn the multi-scale features of the image, but also introduce feature information in both channel and space domain. It enhances the feature representation capability of the network and elevates the effectiveness of the generative model.
ii. We use only a single sample for training, avoiding the problem that the common multi-sample generative models are difficult to fit the real data distribution due to the scarcity of images.
iii. We provide a new data expansion scheme for small sample material images and avoids the overfitting problem of model training in classification tasks. Table 2 in Section 4.2 shows the classification results on subsets of UHCSDB. After data augmentation with generated images, the top-1 accuracy on the test set reaches 95% at the highest.

2. Related work

Data Augmentation in the Material Image Material microstructure image is a kind of texture image. Due to the influence of temperature and different conditions in the thermal processing of materials, the number and shape of crystals in the material microstructure image are highly diverse, and the textures in the image are complicated. This also makes the study of material microscopic images stay in the research phase of observation comparison and artificial statistical analysis. However, the development of materials informatics has changed this situation. The study of material images involves a lot of image processing, and computer vision is an indispensable technology for processing material microscopic images. As with other images, some traditional methods such as geometric transformation or color transformation can be used to augment the material image. Image geometric transformations, including flip, rotation, cropping, deformation, scaling, and other operations. Geometric transformation does not alter the content of the image, it merely selects a portion of the image or redistributes the pixels. Transforming the color space of an image, or adding noise, blurring, random erasing [33], pixel padding, and so on can change the content of the image. For material image datasets with few samples, the use of the above methods has minor effects. Wang et al. [12], when studying the influence of spraying power on YSZ spatter morphology and microstructure of thermal barrier coating, proposed a method using machine vision to automatically identify the lamella in thermal barrier coating and calculate their morphological characteristic parameters. This method converts the color space of the image and uses the median filter to process the image. However, this method can only detect the same class of lamellar images and does not solve the problem of small samples. Ma et al. [13] proposed a data augmentation strategy based on style transfer, which fused simulated images and real images to create composite images and solved the problem of insufficient training data in material microscopic image segmentation. In contrast to our aims and methods, these works all address problems in their respective fields, but do not greatly help in the task of generating microscopic images of small samples of materials.

Images Generative Models In computer vision, images generative model is a significant research direction. Since GAN and VAE were proposed, they have been improved and applied in various fields. The improved GAN models, such as DCGAN [14], BigGAN [15], and ProGAN [16], either solve the problem of GAN training instability to some extent or further improve the quality of generated images. The improved VAE models, such as LogCosh-VAE [17], BNVAE [18], and WAE [19], focus on solving the problems of image blurring or posterior collapse. There are also some works that combine GAN and VAE, such as AAE [20], VAE-GAN [21]. AAE replaces the KL divergence of the posterior distribution in VAE with an adversarial network. The model is trained without reparameterization and performs better than VAE. VAE-GAN jointly trains VAE and GAN and uses a discriminator to measure the similarity of samples, so as to improve the effect of the generative model. However, these improved models rely on a large number of training samples to obtain better generation results. Therefore, single-sample generation is gradually gaining the attention of researchers. Among various single sample generative models, SinGAN [22] can create new object shapes and structures based on preserving the original image patch distribution. ConSinGAN [23] can better maintain the global structure of the image and produce more diverse images than SinGAN. HP-VAE-GAN [10] introduces Patch-VAE [26] and Patch-GAN [34,35] to further generate higher-quality samples. On the one hand, Patch-VAE makes the generated samples have high diversity, and on the other hand, Patch-GAN ensures that the generated samples have finer textures. For material images with very complex textures, the samples generated by HP-VAE-GAN still have some imperfections, such as some textures or noise unrelated to the real image, which needs to be avoided in material image studies.

Attention mechanisms in computer vision Attention mechanism can be categorized into channel attention, spatial attention, and mixed attention. Channel attention focuses on what features are significant, while spatial attention focuses on the location information of features. Attention modules such as SENet (Squeeze-and-Extinction Net) [24], ECA (Efficient Channel Attention) [25], and CBAM (Convolutional Block Attention Module) [11] can play a role in specific tasks. More commonly used attention modules are SENet or CBAM. CBAM combines spatial and channel attention. Overall, CBAM can achieve better results than SENet, which only focuses on channel attention. Fig. 1 illustrates the structure of CBAM: The channel attention module compresses the size of the input feature map F from $C \times H \times W$ to $C \times 1 \times 1$, and then performs element-wise multiplication between the obtained feature map and $F$to obtain the feature map F_c. The spatial attention module compresses the size of the feature map F_c from $C \times H \times W$ to $1 \times H \times W$, and then performs element-wise multiplication between the compressed feature map and F_c to obtain F_{cs}. The process of dimension compression is implemented by global max pooling and global average pooling. The AvgPool (Average Pooling) can effectively learn the range of the target object, and the MaxPool (Max Pooling) can obtain the critical information about the unique target characteristics. The combination of the two methods can infer a better attention map. More details about CBAM can be found in [11]. Our approach integrates CBAM into the model because CBAM can significantly improve the performance of the model with a small number of computations and parameters. Ablation experiments with different attention mechanisms in Section 4.3 also verify the correctness of this choice.

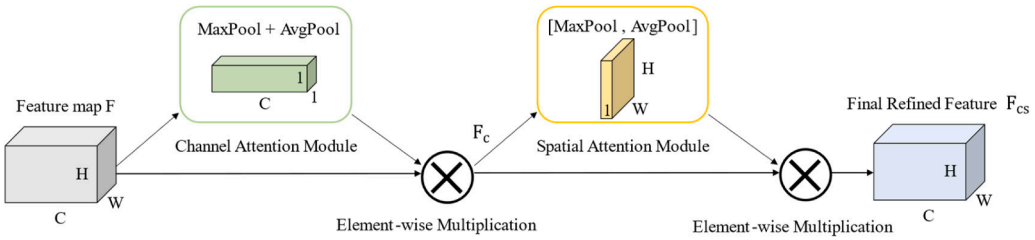


Fig. 1. The Structure of CBAM.

3. Hp-Vae-Gan with CBAM

3.1. Architecture

Fig. 2 shows the improved HP-VAE-GAN model [10], which contains $N+1$ generators, which are denoted as $G^0 \cdots\cdots G^N$ according to the generated results from rough to fine. The model learns the distribution of image patches at different scales from 0to N, and gradually generates images that are close to the true samples from coarse to fine and from low resolution to high resolution. For each scale $n = 0, \cdots\cdots, N$, the training sample $x$is down-sampled to x^n. Different from the original model, we added CBAM to the encoder of Patch-VAE [26] and added a convolutional block to the encoder. The specific structure is shown in Fig. 3. In Patch-VAE, input a single sample x^0, x^0 is cut into patches with the size of r^*r. The sample x^0 is fed to the encoder (CE) that contains CBAM and outputs a feature map $CE(x^0)$ of size H^*W^*C, where H, W, C represent the height, width and channel of the feature map respectively. And the number of the image patches $\rho_j (j = 1 \cdots\cdots T)$ is $T = H^*W$. From

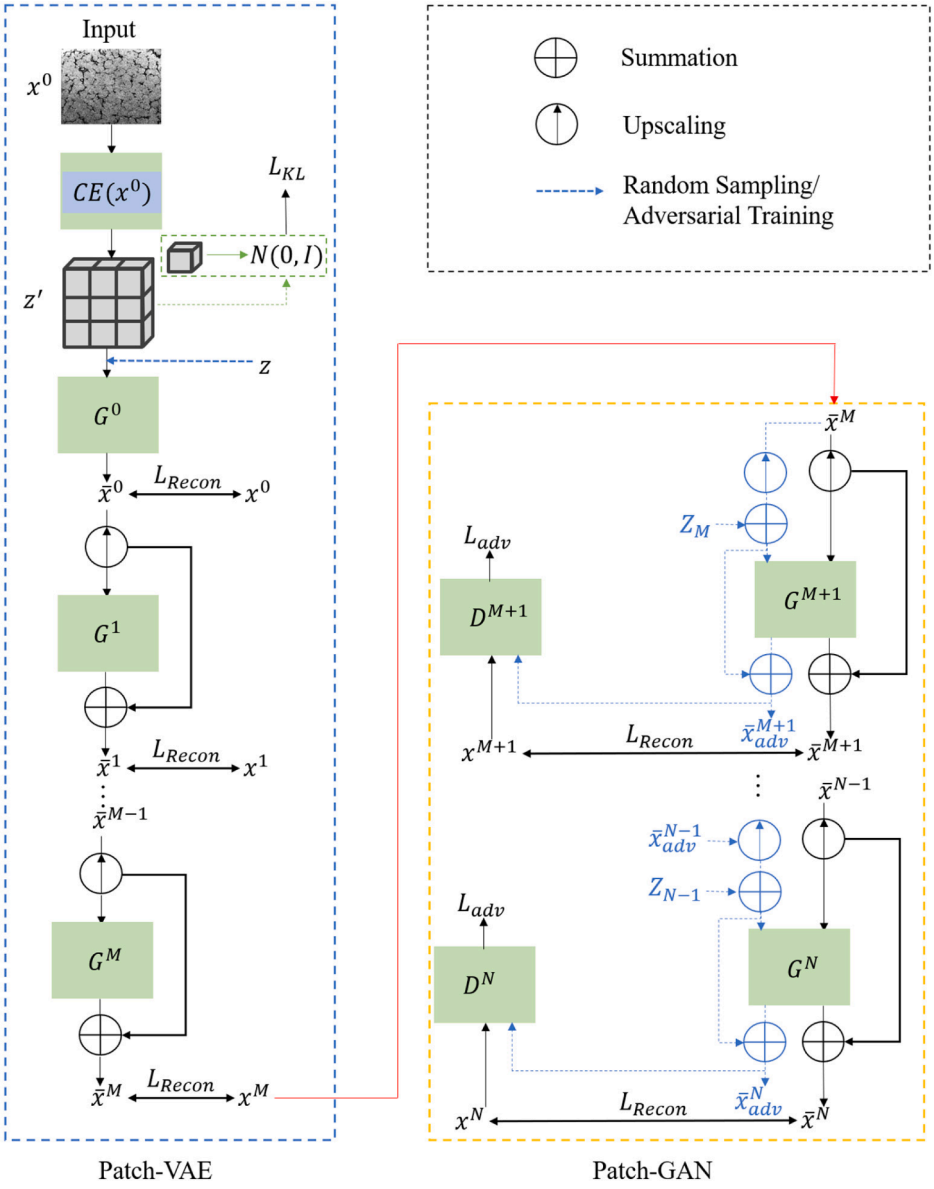


Fig. 2. Overall view of the Improved HP-VAE-GAN. Encoder containing CBAM in Patch-VAE is marked as CE.

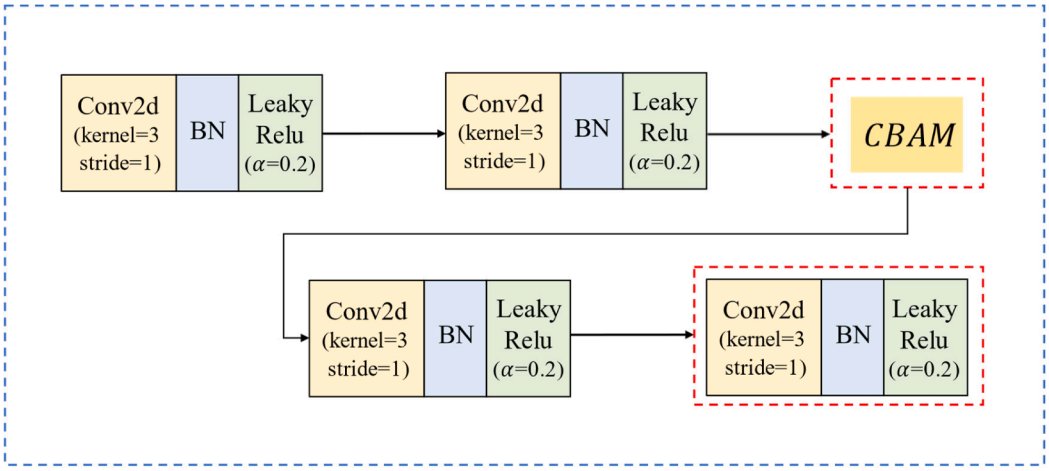


Fig. 3. The structure of improved Encoder. The red dotted line box shows the added CBAM module and convolutional block. The specific structure of CBAM is shown in Fig. 1. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

the perspective of generative model, if the distribution of T patches can be easily fitted, then a great deal of new samples can be generated by directly sampling from this distribution. However, this distribution is difficult to fit directly. Therefore, the model learns from the theory of VAE, hopes to sample latent variable z_j from the prior distribution $P(z_j) = N(0,I)$, and then reconstruct ρ_j from z_j. When the VAE model is implemented, the posterior distribution $P(z_j|\rho_j)$ is assumed to be normal distribution, and the mean μ_j and variance σ_j^2 of $P(z_j|\rho_j)$ are obtained through the fitting of neural network. Then use the reparameterization trick to obtain $z_j^{'} = \epsilon\sigma_j + \mu_j, \epsilon \in N(0,I)$, so as to shift the sampling from $N(\mu_j, \sigma_j^2)$ to $N(0,I)$. In order to align $N(\mu_j, \sigma_j^2)$ with the standard normal distribution, VAE does this by calculating the KL loss of both. The formal expression of the KL loss function is shown in equation (1).

$$L_{KL}(x) = \sum_{j=1}^{T} KL[N(\mu_j, \sigma_j^2) || N(0,I)] \tag{1}$$

The $z^{'}$ in Fig. 2 is the reassembled $z_j^{'}$ obtained by sampling after reparameterization and has the same shape as the feature map $CE(x^0)$. $z^{'}$ is input to G^0 to get the image $\overline{x}^0$, and G^0 is updated through the reconstruction loss of $\overline{x}^0$ and x^0. The image obtained after upsampling $\overline{x}^0$ is then fed into the generator G^1. At this time, the output image of G^1 is added with the upsampled $\overline{x}^0$ to obtain $\overline{x}^1$, and the reconstruction loss of $\overline{x}^1$ and x^1 is calculated again to update encoder G^1. Analogously, until $\overline{x}^M$ is obtained through G^M. The equation [10] is as follows:

$$\overline{x}^n = \uparrow\overline{x}^n + G^n(\uparrow\overline{x}^{n-1}) \tag{2}$$

where $\overline{x}^{n-1}$ is the output of the previous scale, and $\uparrow\overline{x}^{n-1}$ is the result of $\overline{x}^{n-1}$ upsampling to scale n. Before the M scale, VAEs guarantee the diversity of the generated images, making them capable of generating samples with high diversity and not easy to fall into mode collapse. Starting from G^{M+1}, we use a Patch-GAN for each scale, training a generator and a discriminator. The generator adds more detailed textures to samples to ensure the high quality of the generated samples. The discriminator is used to determine if the input is real or fake. When $n > M$, first sample $z \in N(0,I)$ with the same shape as $CE(x^0)$, and then get $\overline{x}^M$ according to equation (2). $\overline{x}^M$ is fed to generator G^{M+1} after upsampling, and the output image by G^{M+1} is added with the upsampled $\overline{x}^M$ to get $\overline{x}^{M+1}$. At the same time, the noise z_M is input, and then it is added with the upsampled $\overline{x}^M$ and sent to G^{M+1}. The output image of G^{M+1} and the input image of G^{M+1} are added to obtain the sample $\overline{x}_{adv}^{M+1}$. Then $\overline{x}_{adv}^{M+1}$ is sent to the discriminator D^{M+1}. By calculating the adversarial loss L_{adv} and reconstruction loss L_{Recon}, updating G^{M+1} and D^{M+1} until the generator G^N outputs finer images. For $n > M$, two different outputs are computed during training. One is $\overline{x}^n$, which is obtained using the recursion of Eq. (2). The other $\overline{x}_{rand}^n$ is obtained using the following equation [10]:

$$\overline{x}_{rand}^n = \begin{cases} \uparrow\overline{x}_{rand}^{n-1} + G^n(\uparrow\overline{x}_{rand}^{n-1} + z_n) n > M \\ \uparrow\overline{x}_{rand}^{n-1} + G^n(\uparrow\overline{x}_{rand}^{n-1}) 0 < n \le M \\ G^0(z^{'}) n = 0 \end{cases} \tag{3}$$

where z_n is the random noise with the same dimension as $\uparrow\overline{x}_{rand}^{n-1}$, and $\overline{x}_{rand}^n$ is the randomly generated sample at scale n. The sample conforms to $\overline{x}_{rand}^n = \gamma x^n + (1-\gamma)\overline{x}_{rand}^n$, where γ is uniformly sampled between 0 and 1. $z^{'}$ is the random noise sampled from the distribution after reparameterization.

3.2. The structure of the improved Patch-Vae

The encoder of HP-VAE-GAN contains three convolutional blocks, each of which consists of a Conv2d convolutional layer, a BN layer, and a Leaky_Relu activation function. The structure of the encoder in the modified Patch-VAE is shown in Fig. 3. We add a convolutional block to the original encoder and insert CBAM in the middle of the four convolutional blocks to augment the extracted features. The functions and parameters used in the network are also shown in Fig. 3. Conv2d is a convolutional operation used to extract different features of two-dimensional input data, where the kernel size is set to 3*3, and the stride is 1. BN (Batch Normalization) layer is used to avoid vanishing gradient. The activation function we choose is Leaky_Relu function with $\alpha = 0.2$.

The decoder still follows the structure of the decoder in HP-VAE-GAN with seven convolutional blocks. Each convolutional block uses the same functions and parameters as the encoder. The deepest and shallowest convolutional block do not contain the BN layer and the Leaky_Relu activation function. The specific decoder structure is shown in Fig. 4.

3.3. Loss function

We continue to follow the loss function of the original HP-VAE-GAN. When the scale $0 < n \le M$, there is no discriminator. The encoder CE is updated by the reconstruction loss $L_{Recon}(\overline{x}^n, x^n)$ of x^n. The loss function under n scale is shown in equation (4) [10].

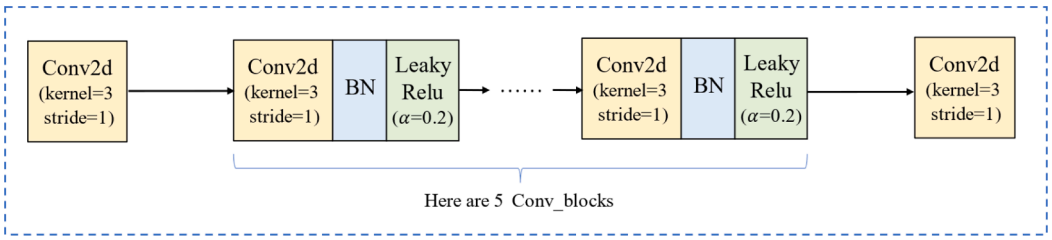


Fig. 4. The structure of Decoder.

$$L_{vae}\left(x^0,\bar{x}^n,x^n\right)=L_{Recon}(\bar{x}^n,x^n)+L_{Recon}\left(\bar{x}^0,x^0\right)+\beta_{vae}L_{KL}\left(x^0\right) \tag{4}$$

The loss is only used to update encoder CE, generator G^0 and G^n. The KL term in the loss function is only related to the encoder CE. Like $\beta-VAE$ [41], the KL item also sets a balance factor β_{vae} as a hyper-parameter. After the discriminator is used, the loss consists of two parts: adversarial loss and reconstruction loss. The reconstruction loss is still $L_{Recon}(\bar{x}^n,x^n)$. The loss $L_{adv}(z,x^n)$ of WGAN-GP is selected for the adversarial loss, where $z\in N(0,I)$. Refer to [27] for the specific content of WGAN-GP. Therefore, when $n>M$, the overall loss function is as follows:

$$L_{adv}(z,\bar{x}^n,x^n)=L_{Recon}(\bar{x}^n,x^n)+\beta_{adv}L_{adv}(z,x^n) \tag{5}$$

In Patch-GAN training, for $n>M$, only G^n and D^n were trained, CE and G^0, … G^{n-1} was frozen.

4. Experiment

4.1. Datasets and Experimental setup

Datasets The experiments are run on a subset of UHCSDB (Ultrahigh Carbon Steel micrograph DataBase) [28]. UHCSDB contains 961 SEM micrographs of UHCS (Ultrahigh Carbon Steel) with size 645*484 and consists of 5 categories: carbide network, pearlite, pearlite + spheroid, spheroid and martensite. We randomly selected 10 micrographs from each category for a total of 50 images, and the ratio of training set to test set is 8:2. The training set is used for image generation and classification model training, and the test set is used to evaluate the results of the classification task. In addition, we also selected several images from the Kylberg texture dataset [39] and STex-512 texture dataset [40] for generation to observe the effect of the model.

Training details For reliable comparison experiments, we adopt the same training settings as HP-VAE-GAN, i.e., $N=9, M=3, r=11$. All images in the dataset were trained using the original size, and the images were augmented by horizontal flipping. The model trains 5000 iterations and Batch_size is 2, and the Adam optimizer with a learning rate of $5*10^{-4}$ is used on each scale. The improved encoder block in the model contains four convolutional blocks and a CBAM module, and the encoder has seven convolutional blocks. The specific parameter settings can be obtained from Fig. 3 and Fig. 4. We use HP-VAE-GAN as a baseline with which our improved model is compared.

4.2. Results and analysis

Qualitative analysis For the images in the subset of UHCSDB, we train the selected training samples one by one, and the number of generated samples for each real sample after training can be set manually. The second row of Fig. 5 shows several randomly chosen generated images. The results show that the steel microscopic images generated by the improved HP-VAE-GAN can clearly display various textures. While the original HP-VAE-GAN generate images with more imperfections, although the texture is also clear. In contrast, the samples generated by our improved model have fewer flaws. By observing the 20 generated images corresponding to each real image in Fig. 6, it can be found that the generated results do not exactly replicate the real image. Each generated image is different, but the style is the same as the real image. This indicates that the images generated with the modified HP-VAE-GAN are of high quality and diversity.

Scatter plot is used to visualize the features of the real images and the generated image. We use the GLCM (Gray-level Co-occurrence Matrix) algorithm [29] to extract the features of the image, and then use Correlation and Dissimilarity to represent. The x-axis and y-axis of the

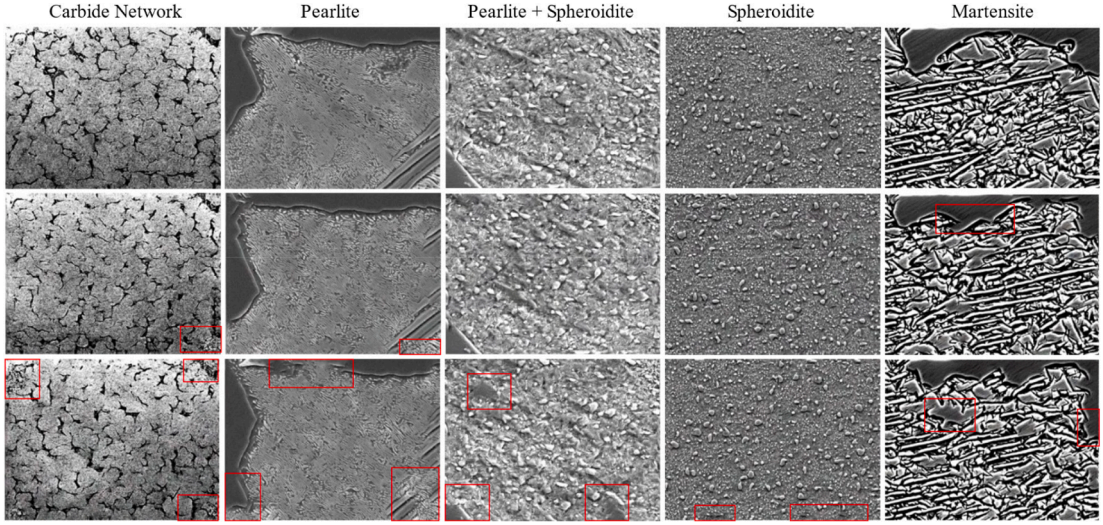


Fig. 5. Images generated after training the model with part of the images in the UHCSDB. The first row shows the real images used for training, the second row shows the images generated using our improved HP-VAE-GAN, and the third row shows the results using HP-VAE-GAN. The red box circles the flaws in the image. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

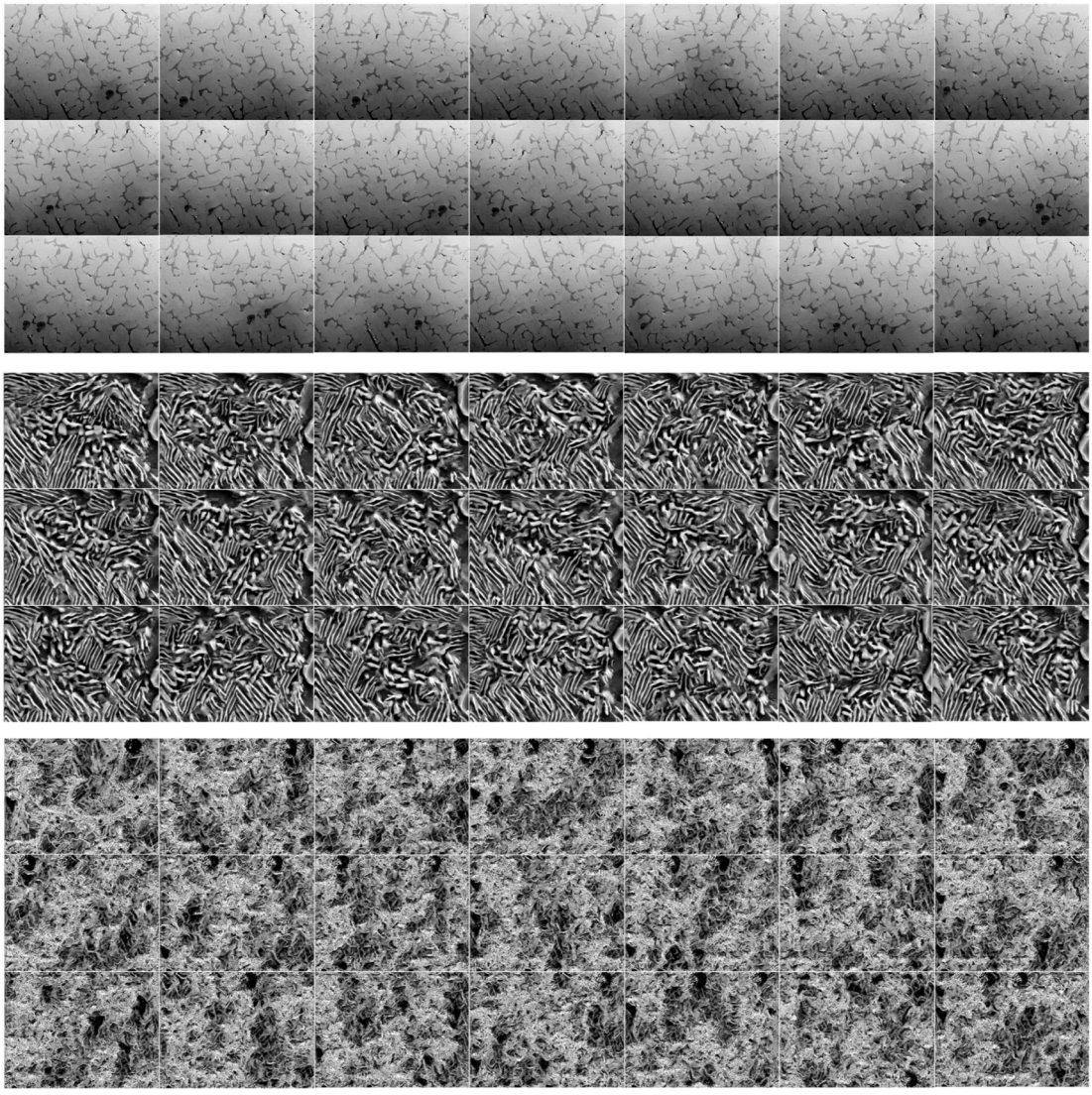


Fig. 6. Each training sample corresponds to 20 generated images. The positions in the first column of the first row, the first column of the fourth row, and the first column of the seventh row are all training samples (real images). The remaining positions are generated images.

scatter plot represent the two features, respectively. Fig. 7 (a) shows the feature scatter plot of 40 real images. There are 5 categories of real images, and each category has 8 images. Fig. 7 (b), Fig. 7 (c), and Fig. 7 (d) respectively show the feature scatter plot of 20, 100, and 500 images corresponding to a real image. It can be seen that when the number of generated samples is large, some samples will be shifted occasionally, but generally, the generated samples are mainly distributed around the real samples.

In addition to generating steel microscopic images, we also selected one image from each of the four categories from the Kylberg dataset and the STex-512 dataset for the experiments. In the Kylberg dataset, four categories are selected: canvas, cushion, linseeds, and stone. In the STex-512 dataset, four categories are selected: bark, fabric, floor, and gravel. Fig. 8 and Fig. 9 show the experimental results using the improved model. The first row is the training samples, and the second row is the generated images. Clearly, the generated images are not simple copies of the real images. There are some differences between the two in texture details. The generated images have strong diversity while preserving authenticity. With the exception of the fabric, the colors of the other generated images are the same as those of their corresponding training samples. The results on different images also validate the effectiveness of the modified HP-VAE-GAN in image generation.

Quantitative analysis Several commonly used metrics, such as Structural Similarity [36], Cosine Similarity [37], Histogram Distance, KL(Kullback-Leibler) Distance and JS(Jensen-Shannon) Distance, are used for quantitative analysis of the generated results. Structural Similarity is an important indicator to evaluate the similarity between two images. It reflects the attributes of texture structure in the image, and its value range is $[-1,1]$. The larger the value of Structural Similarity, the higher the similarity between images. Cosine Similarity is calculated by computing the cosine distance between vectors to represent the similarity of two images. The closer the value is to 1, the more similar the images are. Histogram Distance is a measure of similarity between histograms of two images. Image histogram has rich image details and can reflect the probability distribution of image pixels [38]. Both the KL distance and JS distance are used to measure the difference between two probability distributions. JS distance is a variant of KL distance. The more similar two probability distributions are, the smaller KL distance and JS distance are. KL distance range is $[0, +\infty]$, and JS distance range is $[0,1]$. Table 1 shows the score of image generation using the improved HP-VAE-GAN on the UHCSDB subset, and the scoring results of several metrics show that the proposed method is reliable.

In addition to computing the aforementioned evaluation metrics, we also carry out classification experiments to verify whether augmenting

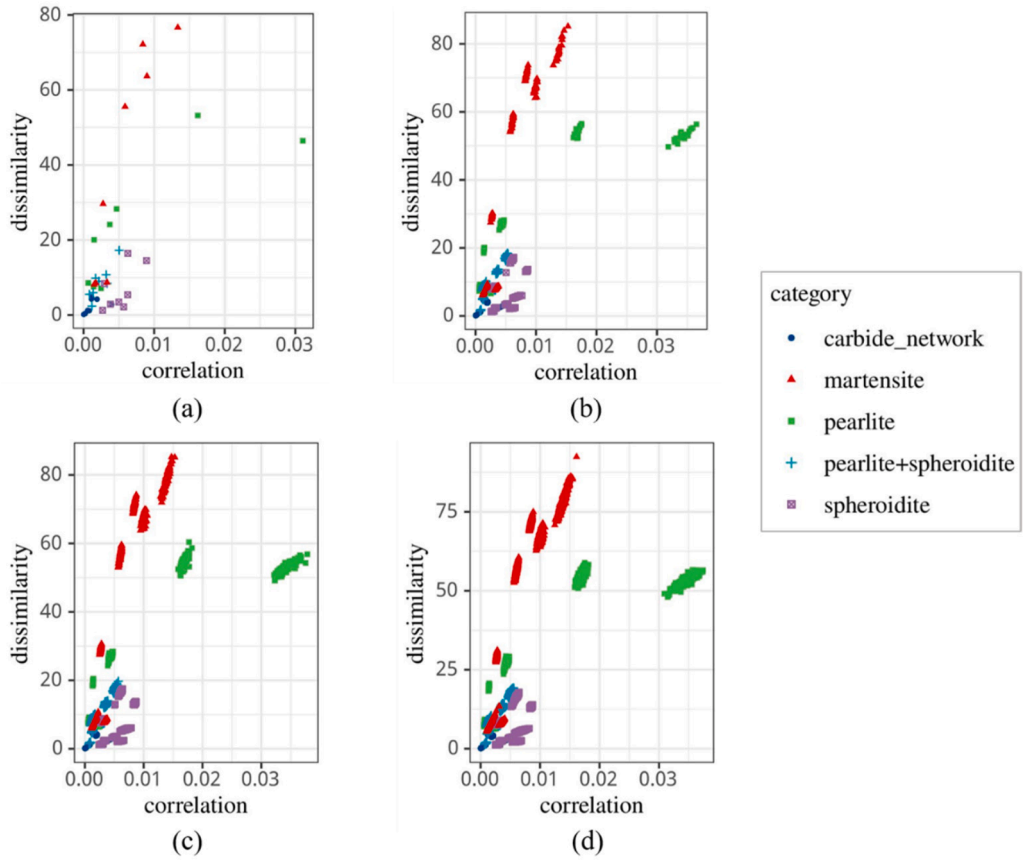


Fig. 7. Scatter plot of real image features and generated image features. (a) is the scatter plot of 40 real images, and (b), (c) and (d) are the scatter plots of 800, 4000 and 20,000 generated images, respectively.

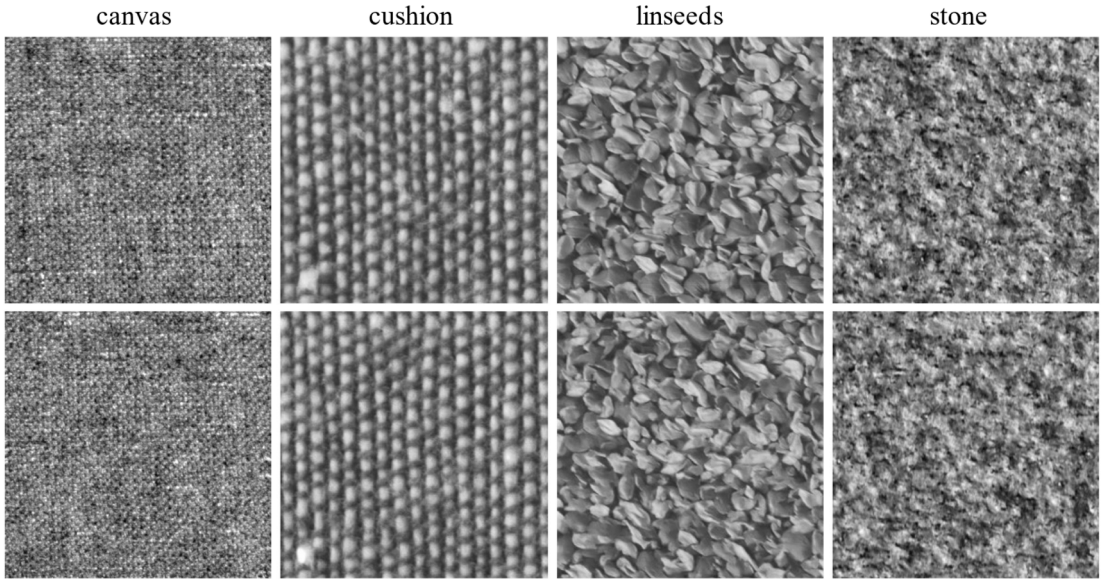


Fig. 8. Experimental results on Kylberg dataset. The first row is the real image for training, and the second row is the image generated using the improved HP-VAE-GAN.

the training set with generated images can increase classification accuracy. The dataset is still a subset of the UHCSDB. In the generation and classification experiments, the division strategy of the dataset is as described in Section 4.1 above, namely, 40 images are used as the training set and 10 images are used as the test set. In the generation experiments, 40 images are trained in turn, and 20 samples of the same size as the real images are generated for each real image, thus the total number of generated samples is 800. For the classification experiments, 40 images are used as the training set and 10 images are used as test set. Because the size of steel microscopic image is too large (645*484), we cut the real image from the middle part to the size of 448*448, and then cut it into 4 224*224 patches. The same process will be performed for

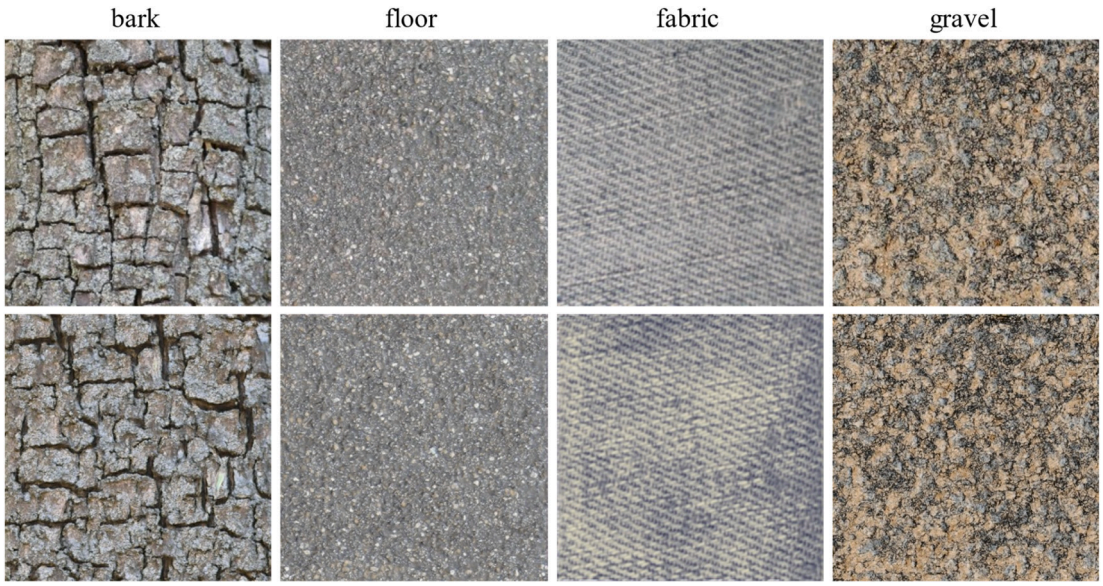


Fig. 9. Experimental results on STex-512 dataset. The first row is the real image for training, and the second row is the image generated using the improved HP-VAE-GAN.

Table 1
The score of metrics.

	Structural Similarity	Cosine Similarity	Histogram Distance	KL Distance	JS Distance
carbide network	0.999899	0.954812	0.895759	0.068004	0.01548
pearlite	0.999866	0.914536	0.552137	0.14235	0.029988
pearlite + spheroidite	0.999823	0.89313	0.62904	0.160105	0.035283
spheroidite	0.99981	0.963096	0.908656	0.041842	0.010241
martensite	0.999875	0.861535	0.543671	0.235227	0.049621

Table 2
Classification results of different models (MobileNet, ResNet50, VGG16) on the test set.

Training set	Test set	Top-1 Accuracy		
		MobileNet	ResNet50	VGG16
Train_A	Test_C	75%	72.5%	40%
Train_B	Test_C	**82.5%**	**90%**	**95%**

Table 3
Classification results of different models in Test_D.

Training set	Test set	Top-1 Accuracy		
		MobileNet	ResNet50	VGG16
Train_A	Test_D	73.41%	67.05%	33.41%
Train_B	Test_D	**81.36%**	**80%**	**77.95%**

the generated samples. Thus, there are 160 (40*4) samples in the training set before expansion, which is recorded as Train_A. The expanded training set adds 3200 (800*4) generated samples based on 160 training samples, a total of 3360 samples, recorded as Train_ B. The test set, denoted Test_C, has 40 (10*4) samples. To avoid accidental results, three classification models, MobileNet [30], ResNet50 [31] and VGG16 [32], were selected for the experiment. The results in Table 2 show that the Top-1 accuracy in the test set increases after training set augmentation using generated images, and the Top-1 accuracy with VGG16 as the classification model increases by 55%. This alleviates to some extent the problem of overfitting of deep learning models and underfitting of target tasks caused by too few training samples. The classification results also demonstrate that the modified HP-VAE-GAN is effective for material image augmentation.

To eliminate the contingency of classification results caused by too few test samples, we add some test samples. The number of images in the test set was increased from 10 to 110. Similarly, we cut each test image into four 224*224patches, so that the new test set has 440 (110*4) samples, and the new test set is denoted as Test_D. Table 3 shows the results after changing the test set. Clearly, the decrease in accuracy compared to the results on Test_C is due to the increase in the number of samples in the test set. Nonetheless, the augmented training set still improves the performance of the classification model compared to the training set without augmentation.

4.3. Ablation experiment

This paper mainly adds CBAM and a convolutional block to the encoder part of Patch-VAE in HP-VAE-GAN. In order to verify the effectiveness of CBAM and convolutional block on the network model, the ablation experiment is carried out. The dataset is still selected as a subset of UHCSDB. In the generation experiment and classification experiment, the partition strategy of the dataset is also as described above. In this experiment, the number of samples generated from one training image is set to 20. The training set of the classification experiment is Train_B, and the test set was Test_C. Table 4 shows the classification results of different models. Model A uses the original HP-VAE-GAN, model B combines CBAM on HP-VAE-GAN, and model C combines CBAM and a convolutional block on HP-VAE-GAN. The experimental results in Table 4 show that after adding CBAM and one convolutional block, HP-VAE-GAN has the most significant improvement in Top-1 accuracy when using VGG-16 to carry out classification experiments, reaching 95%. On 40 test samples, only two samples were

Table 4
Top-1 accuracy of each model. Each row represents different models and shows the classification results on the test set.

	Model	Top-1 Accuracy		
		MobileNet	ResNet50	VGG16
A	HP-VAE-GAN	80%	82.5%	90%
B	HP-VAE-GAN + CBAM	80%	87.5%	**92.5%**
C	HP-VAE-GAN + CBAM + ConvBlock	82.5%	90%	**95%**

predicted incorrectly. The performance of model B with only CBAM is also improved. This also shows that it is feasible to modify the model using CBAM and convolutional blocks.

4.4. Comparison with other attention mechanisms

To compare the effects of different attention mechanisms on the model, this paper also made a comparative experiment of SENet, ECA-Net, and CBAM. Table 5 displays the results of the comparative experiment, and the dataset division strategy is also the same as that in Section 4.3. Model B combines CBAM based on HP-VAE-GAN, Model D combines SENet based on HP-VAE-GAN, and Model E combines ECANet based on HP-VAE-GAN. The insertion position of the attention module is the same, which is behind the second convolutional block of the encoder. The classification results of using VGG-16 in Table 5 show that the effect of adding CBAM is better than SENet or ECANet in HP-VAE-GAN, and the Top-1 accuracy is higher, reaching 92.5%.

5. Conclusion

In this paper, a method of material image augmentation using improved HP-VAE-GAN is proposed. The improved model uses CBAM to refine the feature maps, which enhances the feature representation capability. Meanwhile, a convolutional block is added to the encoder to enhance the feature extraction capability of the network. Experimental results show that the modified HP-VAE-GAN is able to generate high-quality images. The results of the classification experiments show that this approach achieves better results than using HP-VAE-GAN to augment the training set.

Although some achievements have been made in this paper, the model needs further refinement. For example, there is some color inconsistency between the generated results and the training samples, which needs to be removed in the following work. Currently, the proposed method is only applied to the classification of material images. In the future, this approach can be extended to the domain of material image segmentation. Moreover, while this paper focuses on material images, image generation for different datasets, such as medical images, can be considered later to address the small sample issue in these specific domains.

6. Data availability

The data and code used to support this study are available from the GitHub repository https://github.com/thinker-coder/Improved-HP-VAE-GAN.

CRediT authorship contribution statement

Yuexing Han: Conceptualization, Formal analysis, Funding acquisition, Supervision, Writing – review & editing, Methodology. **Yuhong Liu:** Writing – original draft, Visualization, Data curation, Investigation, Methodology. **Qiaochuan Chen:** Formal analysis, Supervision, Writing – review & editing, Methodology.

Table 5
Comparison of different attention mechanisms. Each row represents a different model and shows the classification results on the test set.

	Model	Top-1 Accuracy		
		MobileNet	ResNet50	VGG16
B	HP-VAE-GAN + CBAM	80%	87.5%	**92.5%**
D	HP-VAE-GAN + SENet	82.5%	85%	87.5%
E	HP-VAE-GAN + ECANet	85%	90%	87.5%

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.

Data availability

I have shared the link to my data/code at the "Data availability" of the manuscript.

Acknowledgements

National Natural Science Foundation of China (No. 52273228), Natural Science Foundation of Shanghai (Grant No. 20ZR1419000), Key Research Project of Zhejiang Laboratory (No. 2021PE0AC02), Shanghai Science and Technology Young Talents Sailing Program (No. 23YF1412900).

References

[1] K.T. Butler, D.W. Davies, H. Cartwright, et al., Machine learning for molecular and materials science, Nature 559 (7715) (2018) 547–555.
[2] Yang S, Xiao W, Zhang M, et al. Image Data Augmentation for Deep Learning: a Survey. arXiv e-prints, 2022.
[3] C. Shorten, T.M. Khoshgoftaar, A survey on image data augmentation for deep learning, J. Big Data 6 (1) (2019) 1–48.
[4] I. Goodfellow, J. Pouget-Abadie, M. Mirza, et al., Generative adversarial networks, Commun. ACM 63 (11) (2020) 139–144.
[5] J. Gui, Z. Sun, Y. Wen, et al., A review on generative adversarial networks: algorithms, theory, and applications, IEEE Trans. Knowl. Data Eng. (2021).
[6] Kingma D P, Welling M. Auto-encoding variational bayes. arXiv preprint arXiv: 1312.6114, 2013.
[7] N.V. Chawla, K.W. Bowyer, L.O. Hall, et al., SMOTE: synthetic minority over-sampling technique, J. Artif. Intell. Res. 16 (2002) 321–357.
[8] Inoue H. Data augmentation by pairing samples for images classification. arXiv preprint arXiv:1801.02929, 2018.
[9] Zhang H, Cisse M, Dauphin Y N, et al. mixup: Beyond empirical risk minimization. arXiv preprint arXiv:1710.09412, 2017.
[10] S. Gur, S. Benaim, L. Wolf, Hierarchical patch vae-gan: Generating diverse videos from a single sample, Adv. Neural Inf. Proces. Syst. 33 (2020) 16761–16772.
[11] Woo S, Park J, Lee J Y, et al. Cbam: Convolutionalal block attention module[C]// Proceedings of the European conference on computer vision (ECCV). 2018: 3-19.
[12] Y. Wang, Y. Han, C. Lin, et al., Effect of spraying power on the morphology of YSZ splat and micro-structure of thermal barrier coating, Ceram. Int. 47 (13) (2021) 18956–18963.
[13] B. Ma, X. Wei, C. Liu, et al., Data augmentation in microscopic images for material data mining, npj Comput. Mater. 6 (1) (2020) 1–9.
[14] Radford A, Metz L, Chintala S. Unsupervised representation learning with deep convolutionalal generative adversarial networks. arXiv preprint arXiv:1511.06434, 2015.
[15] Brock A, Donahue J, Simonyan K. Large scale GAN training for high fidelity natural image synthesis. arXiv preprint arXiv:1809.11096, 2018.
[16] Karras T, Aila T, Laine S, et al. Progressive growing of gans for improved quality, stability, and variation. arXiv preprint arXiv:1710.10196, 2017.
[17] Chen P, Chen G, Zhang S. Log hyperbolic cosine loss improves variational auto-encoder. 2018.
[18] Zhu Q, Su J, Bi W, et al. A batch normalized inference network keeps the kl vanishing away. arXiv preprint arXiv:2004.12585, 2020.
[19] Tolstikhin I, Bousquet O, Gelly S, et al. Wasserstein auto-encoders[J]. arXiv preprint arXiv:1711.01558, 2017.
[20] Makhzani, A., Shlens, J., Jaitly, N., Goodfellow, I., & Frey, B. (2015). Adversarial autoencoders. arXiv preprint arXiv:1511.05644.
[21] A.B.L. Larsen, S.K. Sønderby, H. Larochelle, et al., Autoencoding beyond pixels using a learned similarity metric[C]//International conference on machine learning, PMLR (2016) 1558–1566.

[22] T.R. Shaham, T. Dekel, T. Michaeli, Singan: Learning a generative model from a single natural image[C]//Proceedings of the IEEE/CVF, Int. Conference on Computer Vision. (2019:) 4570–4580.
[23] Hinz T, Fisher M, Wang O, et al. Improved techniques for training single-image gans[C]//Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision. 2021: 1300-1309.
[24] Hu J, Shen L, Sun G. Squeeze-and-excitation networks[C]//Proceedings of the IEEE conference on computer vision and pattern recognition. 2018: 7132-7141.
[25] Wang Q, Wu B, Zhu P, et al. ECA-Net: Efficient channel attention for deep convolutionalal neural networks[C]//Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. 2020: 11534-11542.
[26] Gupta K, Singh S, Shrivastava A. Patchvae: Learning local latent codes for recognition[C]//Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. 2020: 4746-4755.
[27] I. Gulrajani, F. Ahmed, M. Arjovsky, et al., Improved training of wasserstein gans, Adv. Neural Inf. Proces. Syst. 30 (2017).
[28] B.L. DeCost, M.D. Hecht, T. Francis, et al., UHCSDB: ultrahigh carbon steel micrograph database, Integr Mater Manuf. Innov. 6 (2017) 197–205, https://doi.org/10.1007/s40192-017-0097-0.
[29] Gadkari D. Image quality analysis using GLCM. 2004.
[30] Howard A G, Zhu M, Chen B, et al. Mobilenets: Efficient convolutionalal neural networks for mobile vision applications. arXiv preprint arXiv:1704.04861, 2017.
[31] He K, Zhang X, Ren S, et al. Deep residual learning for image recognition[C]// Proceedings of the IEEE conference on computer vision and pattern recognition. 2016: 770-778.
[32] Simonyan K, Zisserman A. Very deep convolutionalal networks for large-scale image recognition. arXiv preprint arXiv:1409.1556, 2014.
[33] Z. Zhong, L. Zheng, G. Kang, et al., Random erasing data augmentation[C]// Proceedings of the AAAI conference on artificial intelligence. 34 (07) (2020) 13001–13008.
[34] Demir U, Unal G. Patch-Based Image Inpainting with Generative Adversarial Networks. 2018.
[35] Zhu J Y, Park T, Isola P, et al. Unpaired image-to-image translation using cycle-consistent adversarial networks[C]//Proceedings of the IEEE international conference on computer vision. 2017: 2223-2232.
[36] Wang, Zhou; Bovik, A.C.; Sheikh, H.R.; Simoncelli, E.P.Image quality assessment: from error visibility to structural similarity:IEEE Transactions on Image Processing 13 4 2004-04-01:600–612.
[37] P.N. Tan, M. Steinbach, V. Kumar, Introduction to data mining[M], Pearson Education India, 2016.
[38] F. Camastra, Book review: image processing: principles and applications by Tinku Acharaya, Ajoy K, Ray, 2007.
[39] G. Kylberg. The Kylberg Texture Dataset v. 1.0, Centre for Image Analysis, Swedish University of Agricultural Sciences and Uppsala University, External report (Blue series) No. 35.
[40] Hofbauer Heinz, Huber Stefan. Salzburg Texture Image Database (STex) [DB/OL]. https://wavelab.at/sources/STex/, 2009-5-15/2023-2-16.
[41] Higgins I, Matthey L, Pal A, et al. beta-vae: Learning basic visual concepts with a constrained variational framework[C]//International conference on learning representations. 2017.