Partial Texture VAE: Color and Texture Encoder for Rock Particle Images

Supplementary Material

Tetsushi Yamada
SLB Schlumberger-Doll Research
Cambridge, Massachusetts, USA
tyamada@slb.com

Simone Di Santo
SLB Schlumberger Dhahran Carbonate Research
Dhahran, Saudi Arabia
ssanto@slb.com

A. Perceptual Loss

When deep features are used as a loss function, two types of metrics are typically considered: style loss and content loss [6]. We let $F^l = \phi_l(x) \in \mathbb{R}^{H_l \times W_l \times C_l}$ be the activation map of the l-th layer of a convolutional neural network (CNN) ϕ when processing the image x. Likewise, we let $\hat{F}^l$ the activation map from the image $\hat{x}$ using the same network ϕ. The network ϕ and its layer(s) l to be used are a design decision. While the de facto standard for l is the first five convolutional layers of the VGG network, Zhang *et al.* [13] demonstrated that other pretrained networks perform similarly well. The style loss is the pixel-wise comparison between the Gram matrices $G^l \in \mathbb{R}^{C_l \times C_l}$ and $\hat{G}^l \in \mathbb{R}^{C_l \times C_l}$ respectively computed from F^l and $\hat{F}^l$:

$$\mathcal{L}^l_{\text{style}}(x, \hat{x}) = \sum_{i=1}^{C_l} \sum_{j=1}^{C_l} (G^l_{i,j} - \hat{G}^l_{i,j})^2. \tag{1}$$

The Gram matrix is the map of the inner products between the 2D slices of the activation map in the channel direction, thus it is a symmetric 2D matrix:

$$G^l_{i,j} = \sum_{h=1}^{H_l} \sum_{w=1}^{W_l} F^l_{h,w,i} F^l_{h,w,j}, \tag{2}$$

and similarly,

$$\hat{G}^l_{i,j} = \sum_{h=1}^{H_l} \sum_{w=1}^{W_l} \hat{F}^l_{h,w,i} \hat{F}^l_{h,w,j}. \tag{3}$$

In practice, the Gram matrices are computed efficiently by vectorizing the first two dimensions (H and W) of the feature map and then taking the inner product. Since the Gram matrix is essentially a correlation between the responses of different channels, $\mathcal{L}^l_{\text{style}}$ is agnostic to spatial information in the input images x and $\hat{x}$. The final style loss is the summation of the style losses of each layer:

$$\mathcal{L}_{\text{style}}(x, \hat{x}) = \sum_{l=1}^{L} \mathcal{L}^l_{\text{style}}(x, \hat{x}). \tag{4}$$

The contents loss is a pixel-wise comparison between deep features:

$$\mathcal{L}_{\text{content}}(x, \hat{x}) = \sum_{l=1}^{L} \mathcal{L}^l_{\text{content}}(x, \hat{x}), \tag{5}$$

where

$$\mathcal{L}^l_{\text{content}}(x, \hat{x}) = \sum_{h=1}^{H_l} \sum_{w=1}^{W_l} \sum_{c=1}^{C_l} (F^l_{h,w,c} - \hat{F}^l_{h,w,c})^2. \tag{6}$$

Since the spatial information of the input image is preserved in the activation maps (*e.g.*, [4, 7]), the content loss is considered a structural loss. Gatys *et al.* [5] used the style loss for texture synthesis. Gatys *et al.* [6] and Johnson *et al.* [7] used a combination of the style loss and the content loss for neural style transfer. The learned perceptual image patch similarity (LPIPS) metric [13] is based on the content loss, but learnable weight is applied to F_l for each l. The deep image structure and texture similarity (DISTS) metric [4] introduced a more sophisticated approach, inspired by the structural similarity (SSIM) metric [12], by incorporating both content and style terms using the mean and the variance of each feature map in each layer l.

Note that, to further explore the style loss, we also tested other pooling techniques in the VGG network instead of the default max pooling layer: both the average pooling [5, 6] and weighted L2 pooling [4] that is expected to help avoid aliasing artifacts both theoretically but did not show improvements in our experiments.

B. Variational Autoencoder

The variational autoencoder (VAE) is a probabilistic generative model that extends variational Bayesian inference to a vanilla autoencoder composed of an encoder and a decoder. The encoder encodes the input data x to a latent representation $z \sim \mathrm{Encoder}(x) = q(z \mid x)$. The latent vector z is sampled from the trained posterior distribution $q(z \mid x)$ and is also called a feature or encoding vector. The second part reconstructs the input image from the latent vector z, $\hat{x} \sim \mathrm{Decoder}(z) = p(x \mid z)$, where $p(x \mid z)$ is the trained posterior distribution of the input image. The encoder can be seen as a feature extractor and decoder a generator. VAE regularizes the encoder by imposing a prior over the latent distribution $p(z)$, which is typically the isotropic multivariate Gaussian distribution $z \sim \mathcal{N}(0, \mathbf{I})$, so the encoded feature z has characteristic of being independent unit Gaussian random variables. This ability to control the distribution of the latent space is an advantage of VAE over a vanilla autoencoder in our application as mentioned in the main text. The loss when training VAE is the summation of the negative expected log likelihood (the reconstruction loss) and a prior regularization term:

$$\mathcal{L}_{\mathrm{VAE}} = -\mathbb{E}_{q(z|x)}\left[\log \frac{p(x \mid z)p(z)}{q(z \mid x)}\right] = \mathcal{L}_{\mathrm{recons}} + \mathcal{L}_{\mathrm{prior}}, \tag{7}$$

where

$$\mathcal{L}_{\mathrm{recons}} = -\mathbb{E}_{q(z|x)}\left[\log p(x \mid z)\right] \tag{8}$$

and

$$\mathcal{L}_{\mathrm{prior}} = D_{\mathrm{KL}}\left(q(z \mid x) \parallel p(z)\right). \tag{9}$$

D_{KL} is the Kullback-Leibler divergence, which measures the difference between the distributions $q(z \mid x)$ and $p(z)$. In practice, the negative log-likelihood term $\mathcal{L}_{\mathrm{recons}}$ is an element-wise measure such as binary cross entropy (BCE) and mean squared error (MSE) determined by the type of data.

C. Loss function

We narrow down the choice of our loss functions to perceptual losses: DISTS, LIPPS (VGG-based), and LPIPS (SqueezeNet-based) losses. We also consider the original style loss [6] and a simplified style loss [4] using the feature maps from the first five convolutional layers of the VGG network. To select a suitable reconstruction loss function in our VAE among the five candidates of the similarity measures, we have conducted a content-based image retrieval test using the datasets that do not contain invalid pixels (Tab. 1). For a given query image, similar images are searched and retrieved from a database, and the goodness of the retrieval is evaluated. We used the standard retrieval metrics: precision at 1 (P@1) and mean average precision (MAP). P@1 is the equivalent of the *k*-nearest neighbors (*k*-NN) classification when *k*=1; thus, the score can be seen as a classification accuracy. We used four texture image datasets: colored and grayscale (original) Brodatz [1], KTH-TIPS2b [2, 10], and our own RockTextureLib64 (See the main text). We selected KTH-TIPS2b and Brodatz among other standard texture datasets based on two criteria for the relevance to our rock particle datasets: (1) texture-filled image as opposed to localized texture, and (2) images taken in a controlled environment. We resized their images to 64×64, which corresponds to the size of our training images. As shown in Tab. 1, the simplified style loss showed the best performance with our RockTextureLib64 dataset. Its scores are also stable across four datasets compared with other similarity measures. A reason of the instable results of LPIPS and DISTS might be their trainable weight parameters that are optimized for specific scale of the features in images larger than 64×64. Based on those observations, we chose the simplified style loss as our reconstruction loss function. Details about the datasets and the texture image retrieval test using the standard size can be found in Section D in this Supplementary Material. Thus, we define our VAE loss as

$$\mathcal{L}_{\mathrm{VAE}} = \mathcal{L}_{\mathrm{style\_simple}} + \beta\mathcal{L}_{\mathrm{prior}} \tag{10}$$

where β is a weighting parameter that helps both the reconstruction loss $\mathcal{L}_{\mathrm{style\_simple}}$ and the regularization term $\mathcal{L}_{\mathrm{prior}}$ smoothly decrease during training.

D. Datasets for Image Retrieval Test

We describe further details of the texture image dataset used in the image retrieval test in Section C in this Supplementary Material.

D.1. Brodatz

The dataset originally contains a total of 112 images of different textures with the consistent size of 640×640 (Fig. 1). One image shows one type of texture, and the texture is mostly homogeneous with a few exceptions. We follow the procedure in [4] for the Brodatz datasets in both precision at 1 (P@1) and mean average precision (MAP) computations: we cropped each image into nine non-overlapping patch images with the size of 213×213, which generates a total of 1008 images. For the P@1 test, we used five patches for the database and two for query from each texture (a total of 112 × 5 = 560 patches and 112 × 2 = 224 patches for queries and database, respectively). Since [4] uses the remaining two patches to select *k*, our test is a constrained version of their test (i.e., more challenging), but we believe the results are comparable with their results. For the MAP test, we used three patches for queries, and the remaining six patches as database (a total of 112 × 3 = 336 and 112 × 6 = 732 for queries and

Table 1. Results of texture image retrieval test comparing five texture similarity measures. The sizes of images are 64×64 for all the datasets. The scores are the average of more than one test, and the standard deviations are shown in the parentheses. Two top scores in each column are marked in bold.

	Colored Brodatz (64×64 resized)		Grayscale Brodatz (64×64 resized)		KTH-TIPS2b (64×64 resized)		RockTextureLib64 (64×64 center crop)	
	P@1	MAP	P@1	MAP	P@1	MAP	P@1	MAP
LPIPS (VGG) [13]	0.823 (0.016)	0.675 (0.005)	**0.987 (0.008)**	**0.941 (0.003)**	0.670 (0.032)	**0.579 (0.023)**	0.533	0.065
LPIPS (Squeeze net-based) [13]	0.983 (0.008)	0.931 (0.005)	0.698 (0.019)	0.540 (0.006)	0.620 (0.054)	0.524 (0.020)	0.606	**0.324**
DISTS [4]	**0.998 (0.003)**	**0.971 (0.003)**	0.896 (0.020)	0.762 (0.005)	0.695 (0.023)	0.572 (0.025)	**0.648**	**0.332**
Style loss [5]	**0.992 (0.004)**	0.956 (0.004)	0.923 (0.013)	0.762 (0.004)	**0.697 (0.017)**	0.496 (0.018)	0.589	0.271
Simplified style loss [4]	**0.992 (0.003)**	**0.965 (0.005)**	**0.938 (0.008)**	**0.820 (0.005)**	**0.739 (0.013)**	**0.580 (0.018)**	**0.663**	0.321

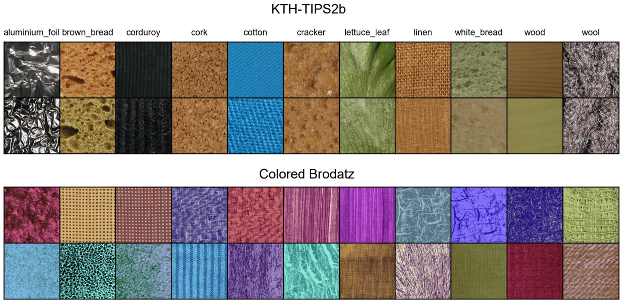

Figure 1. Examples of texture images from the KTH-TIPS2b and Colored Brodatz datasets. From the KTH-TIPS2b dataset, two randomly selected examples from 11 categories are shown. From the Colored Brodatz dataset, 22 randomly selected examples are shown.

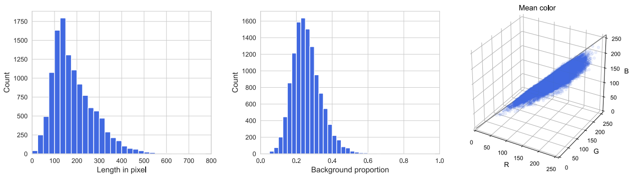

Figure 2. Basic statistics of the rock particle images from RockTextureLib (N=12055). Left: Size (heights and widths combined) distribution of rock particle images. Middle: Proportion of the background pixels (white area) in the image. On average, 25.8% of the pixels is the invalid background. Right: Mean color.

database, respectively). In both cases, the images used for database and for query are randomly selected, and the test is repeated 10 times with different random seeds. We use both colored and grayscale (original) versions of the Brodatz datasets. Recent image classification techniques have good score with Brodatz dataset (e.g., [9]); thus we treat this dataset as relatively easy case of the image retrieval test.

D.2. KTH-TIPS2b

The KTH-TIPS2b dataset is composed of 4752 images from 11 material categories (Fig. 1). Each material category is represented by four samples, and 108 images are taken per sample with all the combinations of three poses, four illuminations, and nine scales. Among 4752 images, 4509 images (94.9%) have the size of 200×200. In the previous works, the images are typically resized to a fixed size and used for evaluation. In our case, we resize all the images to 64×64. We follow the convention and use the standard four train-test split (e.g., [2, 3]) that is predefined to compute P@1 and MAP. One set of images is used for training (database) and three other sets are used for testing (query). Evaluation is run on four possible combinations of sets, and final score is the average of the four scores. This is a more challenging dataset, since some samples of different categories are visually similar and challenging, such as wool vs. cotton (e.g., [11]).

D.3. RockTextureLib

The rock particle instance images in our RockTextureLib dataset vary in size (Fig. 2, left). The H×W size of the smallest instance in terms of pixel area is 18×19 and the largest one is 703×504. The rock particle instance images always contain background pixels at the corners in the image (Fig. 2, middle). On average, about 25.8% of the pixels are background, and the proportion of the background pixels vary depending on the shape of the rock particle. The color of most of the rock particle are close to the grayscale line connecting 0 to 255, where R, G, B have the same values (Fig. 2, right).

While the Brodatz and KTH-TIPS2b datasets do not contain any background pixels, the rock particle dataset does. This prevents a fair comparison of the tested five similarity measures (see the main text) since they are not designed to ignore background pixels. Accordingly, we selected 6595 rock particle images, composed of 1319 images for each of five classes that are large enough and used the 64×64 center-crop images that do not contain background pixels. This dataset is referred to as RockTextureLib64 in the main text, and we used the 64×64 center crop of the RockTextureLib64 images in the retrieval test.

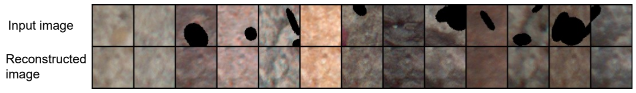

Figure 3. The input image (top row) and the corresponding reconstructed image (bottom row).

E. Image Retrieval Test (Standard Image Size)

We used the 64×64 resized images from Brodatz and KTH-TIPS2b dataset and found that the simplified style loss performed well in the main body. To confirm the superiority of the simplified style loss, we conducted the same experiment with the standard image size. Table 2 shows our results of the texture retrieval test using the standard image size. DISTS and the simplified style loss provides the best results across most cases. Notably, their scores do not drop from the colored version to the original grayscale version of the Brodatz datasets, suggesting that the texture information is well captured in the similarity computation. The results of LPIPS (VGG based) and DISTS are comparable with the same experiment in [4] using Brodatz datasets.

F. Training Details

The training was carried out by optimizing the loss function described in the main text to learn the parameters in the encoder, decoder, and the texture codebook. The balancing weight β was tested and set to 10^{-7}. The parameters of the VGG network used in the simplified style loss were fixed during training. The training was carried out for 5000 epochs with the batch size of 256. Training for one epoch took about 80 seconds with an NVIDIA RTX A5500 GPU. The Adam optimizer with the learning rate of 10^{-4} was used during the entire training. The scheduler for the learning rate of the optimizer did not improve the results. As mentioned the main text, all the 54759 images in the legacy well dataset were used as a training dataset. The training data is split into training and validation datasets with the ratio of 0.8:0.2. The validation dataset was used to monitor the losses to detect the overfitting. The decrease of the $\mathcal{L}_{\text{style\_simple}}$ became negligible and the $\mathcal{L}_{\text{prior}}$ started to increase at epoch 3500, thus we use the model trained up to epoch 3500. Our model was developed in PyTorch. The inference time of the trained model per an instance image was 9.6×10^{-4} seconds with GeForce RTX 3060 GPU and 7.15×10^{-3} seconds with an 8 core CPU.

The images generated from the decoder showed good visual quality capturing the color and texture information of the input images (Fig. 3). The artificially added holes are safely ignored. The perceptual loss is known to generate checkerboard artifacts in some cases [8], but this was not the case with our model.

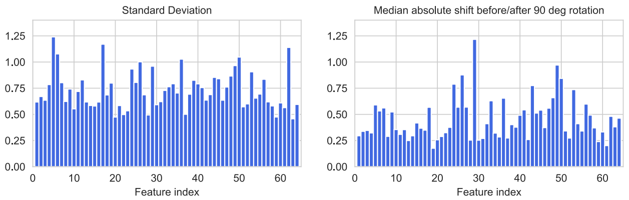

Figure 4. Left: Standard deviation of the features. Right: Median absolute value shift before and after 90 degrees rotation.

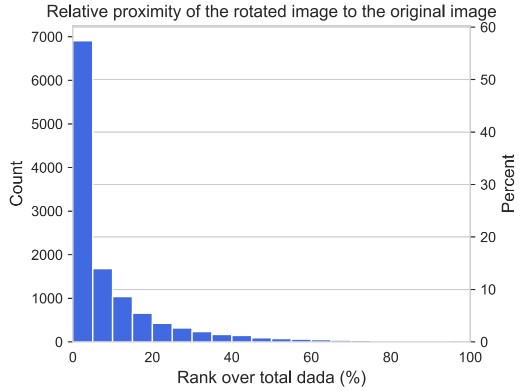

Figure 5. Relative feature proximity between two images: an image and its rotated version by 90 degrees. The proximity (how they are close to each other) is measured by the percentage computed by the similarity rank divided by the total number of data (12055). Each of 12055 images are sampled. The smaller the percentage, the more invariant the features are.

G. Rotation Invariance Analysis

The extracted features from rock particles are ideally rotation invariant: the feature values of the rotated version of the image should be exactly the same as the ones of the original image. This is a challenging topic due to the nature of the operation of the convolution. We observed that the median change of the feature values after rotating the images by 90 degrees was kept low compared with the standard deviation of each feature (Fig. 4). To evaluate it more relatively, we checked if the original image is in the neighborhood of the rotated version of the image in the data space (Fig. 5). It showed that in more than half of the cases, the rotated images are in the proximity of the original images are in the 5% of all the data. Although convolution kernels are not designed nor trained to provide rotation invariance, the rotation invariance somewhat exists, probably thanks to the perceptual losses that is used during training.

References

[1] Safia Abdelmounaime and He Dong-Chen. New Brodatz-based image databases for grayscale color and multiband texture analysis. *ISRN Mach. Vis.*, 2013:876386, 2013. 2

Table 2. Results of texture image retrieval test comparing five texture similarity measures. The images sizes are standard sizes: 213×213 for Brodatz (both color and original grayscale). All the images of KTH-TIPS2b was resized to 200×200. The scores are the average of more than one tests, and the standard deviations are shown in the parentheses. Two top scores in each column are marked in bold.

	Colored Brodatz (213×213)		Grayscale Brodatz (213×213)		KTH-TIPS2b (200×200)	
	P@1	MAP	P@1	MAP	P@1	MAP
LPIPS (VGG-based) [13]	0.976 (0.009)	0.946 (0.005)	0.881 (0.015)	0.765 (0.007)	0.730 (0.026)	**0.618 (0.025)**
LPIPS (Squeeze net-based) [13]	0.985 (0.006)	0.958 (0.004)	0.829 (0.016)	0.685 (0.004)	0.649 (0.040)	0.543 (0.017)
DISTS [4]	**0.999 (0.003)**	**0.989 (0.001)**	**0.980 (0.006)**	**0.932 (0.002)**	**0.745 (0.020)**	**0.618 (0.024)**
Style loss [5]	0.997 (0.004)	0.975 (0.003)	0.957 (0.017)	0.828 (0.005)	0.567 (0.031)	0.433 (0.015)
Simplified style loss [4]	**0.998 (0.003)**	**0.983 (0.002)**	**0.979 (0.010)**	**0.922 (0.004)**	**0.754 (0.017)**	0.586 (0.015)

[2] Barbara Caputo, Eric Hayman, and P Mallikarjuna. Class-specific material categorisation. In *Int. Conf. Comput. Vis.*, volume 2, pages 1597–1604, 2005. 2, 3

[3] Mircea Cimpoi, Subhransu Maji, Iasonas Kokkinos, and Andrea Vedaldi. Deep filter banks for texture recognition, description, and segmentation. *Int. J. Comput. Vis.*, 118(1):65–94, 2016. 3

[4] Keyan Ding, Kede Ma, Shiqi Wang, and Eero P. Simoncelli. Image quality assessment: Unifying structure and texture similarity. *IEEE Trans. Pattern Anal. Mach. Intell.*, 44(5):2567–2581, 2022. 1, 2, 3, 4, 5

[5] Leon Gatys, Alexander S Ecker, and Matthias Bethge. Texture synthesis using convolutional neural networks. *Adv. Neural Inform. Process. Syst.*, 28, 2015. 1, 3, 5

[6] Leon A Gatys, Alexander S Ecker, and Matthias Bethge. Image style transfer using convolutional neural networks. In *IEEE Conf. Comput. Vis. Pattern Recog.*, pages 2414–2423, 2016. 1, 2

[7] Justin Johnson, Alexandre Alahi, and Li Fei-Fei. Perceptual losses for real-time style transfer and super-resolution. In *Eur. Conf. Comput. Vis.*, pages 694–711, 2016. 1

[8] Guilin Liu, Fitsum A Reda, Kevin J Shih, Ting-Chun Wang, Andrew Tao, and Bryan Catanzaro. Image inpainting for irregular holes using partial convolutions. In *Eur. Conf. Comput. Vis.*, pages 85–100, 2018. 4

[9] Li Liu, Jie Chen, Paul Fieguth, Guoying Zhao, Rama Chellappa, and Matti Pietikäinen. From BoW to CNN: Two decades of texture representation for texture classification. *Int. J. Comput. Vis.*, 127:74–109, 2019. 3

[10] P Mallikarjuna, Alireza Tavakoli Targhi, Mario Fritz, Eric Hayman, Barbara Caputo, and Jan-Olof Eklundh. The KTH-TIPS2 database. *Computational Vision and Active Perception Laboratory, Stockholm, Sweden*, 11:12, 2006. 2

[11] Yang Song, Fan Zhang, Qing Li, Heng Huang, Lauren J O'Donnell, and Weidong Cai. Locally-transferred Fisher vectors for texture classification. In *Int. Conf. Comput. Vis.*, pages 4912–4920, 2017. 3

[12] Zhou Wang, Alan C Bovik, Hamid R Sheikh, and Eero P Simoncelli. Image quality assessment: from error visibility to structural similarity. *IEEE Trans. Image Process.*, 13(4):600–612, 2004. 1

[13] Richard Zhang, Phillip Isola, Alexei A Efros, Eli Shechtman, and Oliver Wang. The unreasonable effectiveness of deep features as a perceptual metric. In *IEEE Conf. Comput. Vis. Pattern Recog.*, pages 586–595, 2018. 1, 3, 5