Supplementary Materials: An Encoder-Agnostic Weakly Supervised Method For Describing Textures

Shangbo Mao, Deepu Rajan
College of Computing and Data Science, Nanyang Technological University, Singapore
MAOS0003@e.ntu.edu.sg, ASDRajan@ntu.edu.sg

Appendix A. Experiments

Appendix A.1. Datasets and experiment setting

The quantitative and qualitative evaluation of our Tex^2 on describing texture is on the dataset called DTD^2 [7]. DTD^2 manually annotates the images in the DTD [4] dataset with the visual attributes. Overall, DTD^2 contains 5369 images and 24697 descriptions and provides the official train, val, and test splits. Following [7] [8], we examine our performances on its test set, and use Mean Average Precision (MAP), Mean Reciprocal Rank (MRR), Precision at N (P@N), and Recall at N (R@N) as evaluation metrics.

The performances of texture recognition are evaluated on five representative texture/material benchmark datasets. They are FMD [6], DTD [4], MINC-2500 [1], GTOS [10], and GTOS-M [9]. (1) Flickr Material Database (FMD) [1] consists of ten common material categories, each of them contains 100 images. (2) Describable Texture Database (DTD)[2] [4] contains 47 texture categories, each of which is a texture attribute. For each category, they collected 120 images; (3) Materials in Context Database 2500 (MINC-2500) [1] [3] is a large-scale and open dataset which contains 23 real-world material categories. Each category contains 2500 images. (4) Ground Terrain in Outdoor Scenes Dataset (GTOS) [11] has 40 outdoor material categories. The authors state that since the "snow" class contains only 2 samples, they omit it from their experiments. Thus, GTOS contains 39 categories with a total of 32334 images; (5) GTOS-Mobile [9] is the extended version of the GTOS dataset. Their images are collected with a mobile phone. In total, GTOS-Mobile contains 31 categories. DTD, MINC-2500, GTOS, and GTOS-Mobile datasets provided their official train-test splits. As for FMD, we used the same splits as [12]. To compare with recent state-of-the-art methods [13] [3], for all five datasets, we record the accuracy of every split, and report the accuracy $mean \pm s.t.d.$ of all splits based on 5-run statistics (Each run uses different random seed).

All experiments were implemented with PyTorch and ran on one NVIDIA v100 GPU. For DTD^2, DTD, MINC-2500, GTOS, and GTOS-Mobile, We used SGD optimizer with a learning rate of 0.01 for all learnable modules, momentum of 0.9, weight decay of 1×10^{-4}, batch size of 128. Our model is trained for 10 epochs with a cosine annealing learning rate scheduler. We use three as scaling factor s and four as α for these three datasets. We choose the hyperparameters according to the validation set of MINC-2500 and directly apply these hyperparameters to the other three datasets. As for FMD, the data volume is largely different from the datasets mentioned above. So we separate 10 images per category from the training set as the validation set and then select the hyperparameters according to it. In this case, the learning rate of ResNet50 is 0.0001 and the remaining learnable module is 0.001, the training epochs become 20, the batch size is 16, the scaling factor $s = 2$, and $\alpha = 1$.

ResNet50 was initialized with ImageNet-pretrained models. CLIP model was initialized with pretrained model weights downloaded from their official website [4] and kept frozen during the training process in all our experiments. The BERT model used here is called 'bert-base-uncased', and was initialized with pre-trained weights downloaded from Hugging Face website [5], and kept it frozen like the CLIP model. For image preprocessing on the FMD, DTD, and MINC-2500 datasets, we follow the same preprocessing procedures as CLIP models. This involves resizing images to 224×224, center cropping images to 224×224, and then normalizing the images. Since the images from FMD, DTD and MINC-2500 are captured from the everyday scenarios similar to the training images used for CLIP, this preprocessing approach is appropriate. However, the

[1] FMD: https://people.csail.mit.edu/celiu/CVPR2010/FMD/
[2] DTD: https://www.robots.ox.ac.uk/~vgg/data/dtd/download/dtd-r1.0.1.tar.gz
[3] MINC-2500: http://opensurfaces.cs.cornell.edu/publications/minc/
[4] CLIP: https://github.com/OpenAI/CLIP
[5] 'bert-base-uncased': https://huggingface.co/bert-base-uncased

Table A. The impact of texture description on the accuracy of texture recognition

	Image Encoder	Text Encoder	0.0	0.1	0.2	0.3	0.4	0.5
Recog	Res50	–	74.09	71.09	67.98	62.19	54.55	44.04
Recog & Desc	Res50	CLIP	74.21	71.39	67.39	62.58	54.45	44.73

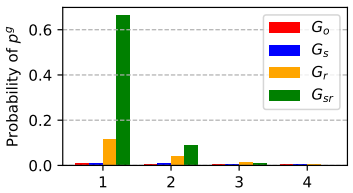


Figure A. Top-4 probability of p^g.

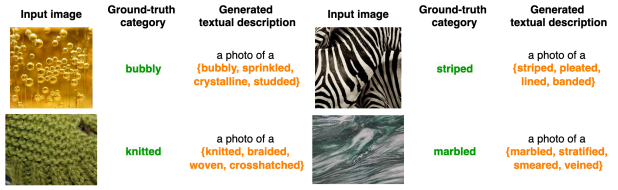


Figure B. Examples of generated texture descriptions in DTD dataset

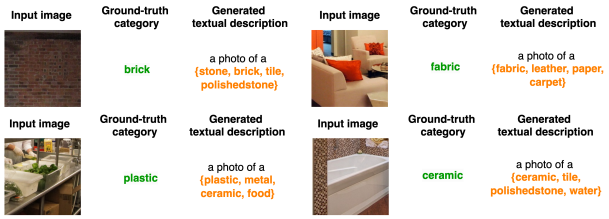
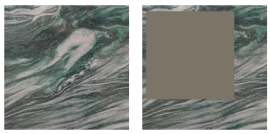


Figure C. Examples of generated texture descriptions in MINC-2500 dataset

GTOS and GTOS-Mobile datasets, consisting of ground terrain images, necessitate a slightly different preprocessing procedure due to their distinct characteristics.

For GTOS, images are resized to 224 × 224, center cropped to 224 × 224, and then normalized based on parameters specified in the GitHub repository of DEP [9]. For GTOS-Mobile, our preprocessing procedure takes cues from both DEP [9] and CLIP [5]. During training, images are resized to 224 × 224, center cropped to 224 × 224, applied color jitter, added lighting noise, and normalized based on the parameters used in CLIP. During testing, we adopt the same preprocessing operations as CLIP on images.



(a) (b)

Figure D. (A) Original image; (B) Randomly masked image (masking ratio=0.5))

Appendix A.2. Analysis of ranked and scaled operations

According to Eq. (a), $\mathcal{L}_{KL}$ imposes a greater penalty when the difference between p^g and p^q is more pronounced.

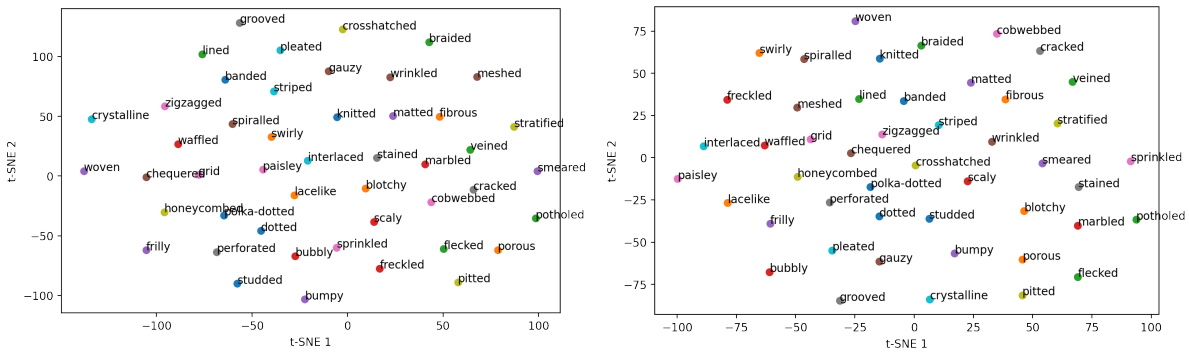
$$\frac{\partial \mathcal{L}_{KL}}{\partial p^q} = \frac{\partial \sum_j p^g \log \frac{p^g}{p^q}}{\partial p^q} = -\sum_j \frac{p^g}{p^q} \tag{a}$$

We show top-4 values of $p^g = softmax(G_{(\cdot)})$ for 'banded' category in DTD in Fig. A. G_o is the cosine similarity between the text embeddings of category and lexicon, as displayed in Eq. 1 of the main manuscript. It is calculated based on the text embeddings extracted by CLIP ('ViT-B/32') using prompt 'An image of $\{\cdot\}$ texture.' with scale=2, rank=3. Due to subtle differences of textures and their attributes, $p^g = softmax(G_o)$ typically appears as an almost flat distribution. It shows that applying the ranked operation makes $p^g = softmax(G_r)$ steeper than $p^g = softmax(G_s)$.

Tab. 2 in the main manuscript also demonstrates that the ranked outperforms the scaled. This confirms that a **sharper** p^g demands $\mathcal{L}_{KL}$ impose a greater penalty on the **most relevant visual attributes** identified by p^g, resulting in improved performance in phrase retrieval.

Appendix A.3. Texture description generated during texture recognition

In this section, we delve into the texture descriptions generated by the WTDG module during the performance evaluation of texture recognition on the DTD and MINC-2500 dataset. The examples illustrated in Fig. B and Fig. C are the output of the model we refer to as Tex^2_{early}, utilizing CLIP with "VIT-L/14" encoder for text encoder and ResNet50 for image encoder. Our WTDG successfully identifies the ground-truth category as the top-1 phrase at the most cases, with all subsequent phrases semantically



(a) (b)

Figure E. (A) t-SNE plot of Recog trained classifiers; (B) t-SNE plot of Recog & Desc trained classifiers

relevant to the input image from the employed lexicon $\mathbb{L}$. Similar to the observations made in Section 4.1, our WTDG excels at selecting the most relevant phrases from the given lexicon, specifically the ground-truth category and other semantically relevant texture attributes. This efficacy underlines the capability of our proposed Tex^2 framework to not only generate accurate texture descriptions but also concurrently recognize the correct categories simultaneously.

Appendix A.4. The effect of texture description on texture recognition

In this section, we analyzed if texture recognition performance will be affected when introducing an additional texture description task into our framework. First, we remove all the elements related to the texture description task i.e., text encoder, WTDG, and image-text fusion to compare with our proposed Tex^2 in Tab. A. According to this table, incorporating an extra task, the performance of texture recognition is enhanced by 0.1%. It means that describing textures cannot harm the texture recognition task but assists it in obtaining better performances.

To verify this statement, we checked the semantic relationship of their trained classifier weights by plotting their t-SNE plot in Fig. E. According to Fig. E(A), texture recognition trained classifiers tend to make visually similar categories locate closer like "knitted" with "braided" and "matted", and "meshed" with "grid". However, in Fig. E(B) by training these two tasks together, "knitted" is not only located close to "braided" but also "woven", and "meshed" closer to "perforated", which consider both appearance and semantic meanings. We can conclude that the rough semantic relationship learned by SR-KL can be distilled into the image encoder and the classifier, which will benefit texture recognition eventually.

When adding masking to the testing images with different area ratios, we find that the two-task trained model has relatively better robustness to retain almost the same or better texture recognition performance with different levels of image degradation i.e., image masking. When we masked 50% image, Tex^2 can perform better by around 0.7%. Fig. D shows the original image and the randomly masked image.

Appendix A.5. Discussion on limitations

The limitations of our proposed method can be discussed in the context of two tasks:
(1) Limitation on texture description: The pseudo-target in the proposed SR-KL loss ensures that at least the top-5 retrieved phrases belong to the ground-truth texture descriptions, as reflected in the promising MRR, P@5, and R@5 results. However, the lower MAP score indicates that WTDG struggles to correctly rank the remaining phrases. The possible reasons are: (a) 71.15% of images in the DTD^2 dataset have their GT category present in the GT texture description. For these images, WTDG effectively retrieves the correct category, resulting in a high MRR. (b) However, the remaining relevant phrases identified by our pseudo-targets may not fully align with the GT descriptions. It is also worth noting, as shown in Fig. 6, that in some cases the GT descriptions do not include all relevant phrases, and some phrases retrieved by WTDG, while highly relevant to the texture image, were not included in the GT descriptions.
(2) Limitation on texture recognition: The performance of Tex^2 is affected by the distinct backgrounds and contexts across different datasets. While the pre-trained text encoder effectively provides meaningful semantic relationship guidance for real-world material classes, it is less effective for domain-specific classes. As a result, Tex^2 performs very well on DTD and MINC-2500 but less effectively on GTOS and GTOS-Mobile. These two limitation will be addressed in our future work.

Appendix A.6. Texture Lexicon

Following is the texture lexicon [2] used in our Tex^2 below:
{'asymmetrical', 'banded', 'barred', 'spattered', 'blemished', 'blotchy', 'braided', 'bubbly', 'bumpy', 'chequred', 'clotted', 'cloudy', 'coarse', 'cobweb', 'coilded', 'complex', 'corkscrew', 'corrugated', 'cracked', 'creased', 'crinkled', 'crosshatched', 'crows', 'crumpled', 'crystalline', 'curly', 'cyclical', 'dense', 'discontinous', 'disordered', 'dotted', 'entwine', 'facetted', 'fibrous', 'filigree', 'fine', 'flecked', 'flowing', 'fractured', 'fragmented', 'freckled', 'fretted', 'frilly', 'frothy', 'furrowed', 'gauzy', 'gouged', 'granular', 'gravelly', 'grid', 'grille', 'gritty', 'grooved', 'harmonious', 'holey', 'honeycombed', 'indefinite', 'interlaced', 'intertwined', 'irregular', 'jumbled', 'kinky', 'knitted', 'knotty', 'lacelike', 'lattice', 'lined', 'marbled', 'matted', 'meshed', 'messy', 'mottled', 'muddy', 'netlike', 'nonuniform', 'oriented', 'patchy', 'patterned', 'pebbly', 'perforated', 'periodic', 'pimpled', 'pitted', 'pleated', 'pockmarked', 'polka', 'porous', 'potholed', 'powdery', 'random', 'regular', 'repetitive', 'rhythmic', 'ribbed', 'riddled', 'ridged', 'rippled', 'rough', 'rocky', 'ruled', 'rumpled', 'sandy', 'scalloped', 'scaly', 'scarred', 'scrambled', 'scratched', 'simple', 'sinewy', 'smeared', 'smooth', 'smudged', 'speckled', 'spiralled', 'spongy', 'spotted', 'sprinkled', 'stained', 'stratified', 'streaked', 'striated', 'stringy', 'striped', 'studded', 'swirly', 'tangled', 'uniform', 'veined', 'waffled', 'wavy', 'webbed', 'well', 'whirly', 'winding', 'wiry', 'wizened', 'woven', 'wrinkled', 'zigzag'}

References

[1] Sean Bell, Paul Upchurch, Noah Snavely, and Kavita Bala. Material recognition in the wild with the materials in context database. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3479–3487, 2015. 1

[2] Nalini Bhushan, A Ravishankar Rao, and Gerald L Lohse. The texture lexicon: Understanding the categorization of visual texture terms and their relationship to texture images. *Cognitive science*, 21(2):219–246, 1997. 4

[3] Zhile Chen, Feng Li, Yuhui Quan, Yong Xu, and Hui Ji. Deep texture recognition via exploiting cross-layer statistical self-similarity. In *Proceedings of the IEEE/CVF conference on Computer Vision and Pattern Recognition*, pages 5231–5240, 2021. 1

[4] Mircea Cimpoi, Subhransu Maji, Iasonas Kokkinos, Sammy Mohamed, and Andrea Vedaldi. Describing textures in the wild. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3606–3613, 2014. 1

[5] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. In *International conference on machine learning*, pages 8748–8763. PMLR, 2021. 2

[6] Lavanya Sharan, Ce Liu, Ruth Rosenholtz, and Edward H Adelson. Recognizing materials using perceptually inspired features. *International journal of computer vision*, 103:348–371, 2013. 1

[7] Chenyun Wu, Mikayla Timm, and Subhransu Maji. Describing textures using natural language. In *European Conference on Computer Vision*, pages 52–70. Springer, 2020. 1

[8] Zelai Xu, Tan Yu, and Ping Li. Texture bert for cross-modal texture image retrieval. In *Proceedings of the 31st ACM International Conference on Information & Knowledge Management*, pages 4610–4614, 2022. 1

[9] Jia Xue, Hang Zhang, and Kristin Dana. Deep texture manifold for ground terrain recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 558–567, 2018. 1, 2

[10] Jia Xue, Hang Zhang, Kristin Dana, and Ko Nishino. Differential angular imaging for material recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 764–773, 2017. 1

[11] Jia Xue, Hang Zhang, Ko Nishino, and Kristin J Dana. Differential viewpoints for ground terrain material recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 44(3):1205–1218, 2020. 1

[12] Wei Zhai, Yang Cao, Jing Zhang, Haiyong Xie, Dacheng Tao, and Zheng-Jun Zha. On exploring multiplicity of primitives and attributes for texture recognition in the wild. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2023. 1

[13] Wei Zhai, Yang Cao, Jing Zhang, and Zheng-Jun Zha. Deep multiple-attribute-perceived network for real-world texture recognition. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 3613–3622, 2019. 1