Multiscale Graph Texture Network Supplementary Material

Ravishankar Evani, Deepu Rajan, and Shangbo Mao

College of Computing and Data Science, Nanyang Technological University, Singapore
{S220007,ASDRajan,MAOS0003}@e.ntu.edu.sg

1 Data Augmentation

Details of data augmentation techniques used in the experiments for Multiscale Graph Texture Network (GTN) are shown in this supporting material. Results in the paper can be reproduced using data augmentations shown in Tab. 1 which are applied to images during training. During the testing phase, following previous work [1], we simply resize input images to 256×256 resolution and crop 224×224 resolution at the center of the image . Details of the augmentations presented in Tab. 1 can be found in PyTorch documentation.

Table 1: Data augmentation used in experiments.

Augmentation Technique	DTD	FMD	KTH-TIPS2-b	GTOS	GTOS-Mobile
Resize + Center Crop	✗	✓	✓	✓	✓
Resize + Random Resized Crop	✓	✗	✗	✗	✗
Random Rotation (5°)	✗	✗	✓	✓	✗
Random Horizontal Flip	✓	✓	✓	✓	✓
Random Vertical Flip	✓	✓	✓	✓	✓
Random Equalize	✗	✓	✓	✓	✓
Random Auto-contrast	✓	✓	✓	✓	✓
Color Jitter	✓	✓	✓	✓	✓

However, it is noteworthy that GTN achieves state-of-the-art results on several datasets using ConvNeXt backbones, even without additional data augmentation. Tab. 2 presents the performance of GTN when no additional data augmentation techniques are used (only perform resizing followed by center cropping or random resized cropping, as specified in Tab. 1 for the respective datasets), during training. GTN accuracy values in bold indicate superiority over the current state of the art, RADAM [2]. The results for RADAM are shown in Table 1 of the paper.

Finally, we conducted a study to understand the effect of removing each augmentation technique on GTN's performance on a texture dataset (DTD) and two material datasets (FMD and KTH). The results are shown in Tab. 3. Removing specific augmentation techniques has very little effect on GTN's performance when trained and evaluated on the DTD and KTH datasets. However, GTN

Table 2: A Comparison of GTN's performance in terms of classification accuracy (%) on texture benchmarks when no additional augmentation techniques are applied. **Bolded** results indicate it is superior to current state-of-the-art RADAM for that specific backbone-dataset combination

Backbone	DTD	FMD	KTH	GTOS	GTOS-M
ConvNeXT-N	**75.4** $\pm$0.8	**87.3** $\pm$2.3	87.7 $\pm$2.8	83.5 $\pm$1.8	**84.4**
ConvNeXT-T	**77.6** $\pm$0.9	86.1 $\pm$3.4	90.2 $\pm$3.9	84.0 $\pm$1.4	**88.0**
ConvNeXT-B	**78.0** $\pm$0.7	87.9 $\pm$2.4	**88.5** $\pm$4.8	**84.5** $\pm$1.7	**89.1**
ConvNeXT-L	**78.7** $\pm$0.7	88.0 $\pm$3.7	**90.7** $\pm$4.2	**85.1** $\pm$1.6	**88.7**

shows some improvement when both random horizontal and vertical flips (each with a 50% probability) are applied to the FMD dataset.

Table 3: A Comparison of GTN's performance in terms of classification accuracy (%) on texture benchmarks when specific augmentation techniques are excluded. ConvNeXt-T is used as backbone.

w/o Augmentation Technique	DTD	FMD	KTH
w/o Random Rotation (5°)	-	-	91.9 $\pm$4.8
w/o Random Horizontal Flip	77.6 $\pm$1.1	86.3 $\pm$4.1	91.8 $\pm$5.1
w/o Random Vertical Flip	77.8 $\pm$0.9	86.1 $\pm$3.3	92.2 $\pm$5.1
w/o Random Equalize	-	86.8 $\pm$3.5	92.0 $\pm$4.8
w/o Random Auto-contrast	77.6 $\pm$1.1	86.6 $\pm$3.8	92.0 $\pm$4.4
w/o Color Jitter	77.7 $\pm$1.0	86.9 $\pm$3.9	92.2 $\pm$4.7

2 Hyperparameters

Tab. 4 shows the hyperparameters used for each of the datasets. We use a batch size of 32 for for all datasets. In equation 4 of the paper, setting $\rho = 1$ implies a sum operation while setting $\rho = |\mathcal{N}(i) \bigcup \{i\}|^{-1}$ implies a mean operation. For this paper, we selected the later to perform moderate adjustments (instead of significant adjustments) to the values in $\mathbf{X}_i^{gm}$. However, significant adjustments to $\mathbf{X}_i^{gm}$ can improve GTN's performance in some cases. As shown in Tab. 5 (with no additional augmentation), setting $\rho = 1$ with ConvNeXT-T/B backbone improves GTN's performance. Conversely, setting $\rho = |\mathcal{N}(i) \bigcup \{i\}|^{-1}$ with ConvNeXT-N/L backbone, enhances GTN's performance.

3 Explanation of Inflation Function

As shown in Eq. (1), the inflation function used in equation 3 in the paper is a sigmoid with a bias of 1. This is to ensure that the lower bound of the re-

Table 4: Training hyperparameters for each dataset

Dataset	Epochs	Weight Decay	Focal Loss
DTD	300	0.5	$\alpha = 1, \gamma = 2$
FMD	100	0.01	$\alpha = 1, \gamma = 1$
KTH	30	0.5	$\alpha = 1, \gamma = 1$
GTOS-M	30	0.01	$\alpha = 1, \gamma = 1$
GTOS	30	0.01	$\alpha = 1, \gamma = 1$

Table 5: A Comparison of GTN's performance in terms of classification accuracy (%) on KTH Dataset when no additional data augmentation techniques are applied, using different values of ρ . The best result for each backbone-ρ combination is **bolded**

Backbone	$\rho = 1$	$\rho = \lvert\mathcal{N}(i) \bigcup \{i\}\rvert^{-1}$
ConvNeXT-N	87.6 $\pm_{2.8}$	**87.7** $\pm_{2.8}$
ConvNeXT-T	**90.4** $\pm_{3.8}$	90.2 $\pm_{3.9}$
ConvNeXT-B	**88.6** $\pm_{4.8}$	88.5 $\pm_{4.8}$
ConvNeXT-L	90.6 $\pm_{4.2}$	**90.7** $\pm_{4.2}$

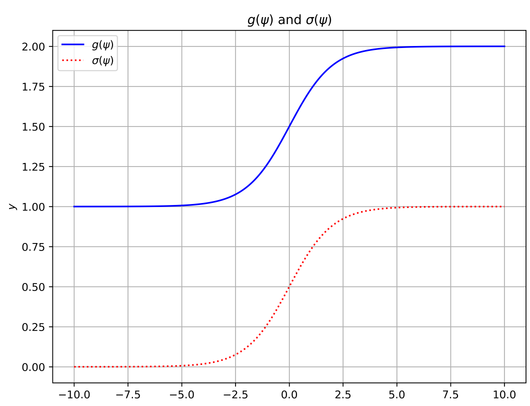

Fig. 1: Illustration comparing original sigmoid function and the inflation function

calibrated self-loop weights are close to it's original value and the upper bound is close to twice it's original value. The extent of inflation is controlled by the learnable parameter, ψ_m. Fig. 1 illustrates the difference between $g(\psi_m)$ and $\sigma(\psi_m)$.

$$
\begin{aligned}
g(\psi_m) &= \frac{2 + e^{-\psi_m}}{1 + e^{-\psi_m}} \\
&= \frac{1 + 1 + e^{-\psi_m}}{1 + e^{-\psi_m}} \\
&= \frac{1}{1 + e^{-\psi_m}} + 1 \\
&= \sigma(\psi_m) + 1
\end{aligned} \tag{1}
$$

References

1. Chen, Z., Li, F., Quan, Y., Xu, Y., Ji, H.: Deep texture recognition via exploiting cross-layer statistical self-similarity. In: 2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). pp. 5227–5236 (2021). https://doi.org/10.1109/CVPR46437.2021.00519 1
2. Scabini, L., Zielinski, K.M., Ribas, L.C., Gonçalves, W.N., De Baets, B., Bruno, O.M.: Radam: Texture recognition through randomized aggregated encoding of deep activation maps. Pattern Recognition **143**, 109802 (2023). https://doi.org/https://doi.org/10.1016/j.patcog.2023.109802, https://www.sciencedirect.com/science/article/pii/S0031320323005009 1