Neural Transformation Network to Generate Diverse Views for Contrastive Learning (Supplementary material)

Taekyung Kim*
KAIST
tkkim93.personal@gmail.com

Debasmit Das
Qualcomm AI Research†
debadas@qti.qualcomm.com

Seokeon Choi
Qualcomm AI Research†
seokchoi@qti.qualcomm.com

Minki Jeong*
KAIST
mkjeong033@gmail.com

Seunghan Yang
Qualcomm AI Research†
seunghan@qti.qualcomm.com

Sungrack Yun
Qualcomm AI Research†
sungrack@qti.qualcomm.com

Changick Kim
KAIST
changick@kaist.ac.kr

1. Algorithms for the proposed method

We provides algorithms for the following stages of the proposed framework: 1) neural style transformation network training, 2) view generation, 3) encoder training, and 4) transfer learning, as shown in Algorithms 1, 2, 3, and 4, respectively.

2. Ablation study on geometric transformation configurations

To search for complementary geometric transformations with our neural transformation framework, we conducted ablation study on several configurations of the geometric transformation. Table 1 shows the encoder pretraining performance comparison results of some configuration candidates. Here, the SimCLR method is applied for contrastive learning, and the rotation transformation is applied 90-degree-wise. We first guess that the random rotation transformation will boost the effectiveness of our method on contrastive learning, but these transformation turned out to interfere the representation learning. Moreover, applying random perspective transformations during our view generation process negatively impacts on contrastive learning. Therefore, we decided not to use random rotation and perspective transformations in our transformation framework.

3. Details on datasets

3.0.1 CIFAR-10

The CIFAR-10 dataset consists of 32x32 images in 10 classes. Each class has 6000 images. The training and test split contains 50000 and 10000 images, respectively.

3.0.2 MSCOCO

The MSCOCO dataset is a large-scale dataset for various tasks including classification, object detection, segmentation, keypoint estimation, dense pose estimation, and captioning. The latest dataset consists of 328k images, but we use the past version of the dataset released in 2017, which consists of 118k training images and 5k validation images.

3.0.3 Aircraft

Fine-Grained Visual Classification of Aircraft, which we call Aircraft in short, is a benchmark dataset released for the fine-grained image classification of aircraft. The dataset consists of 10,200 images with 100 images for each of 102 different aircraft model variants.

3.0.4 DTD

The Describable Textures Dataset (DTD) is a collection of textural images in the wild consisting of 5640 images, 120 images for 47 categories each. The labels are annotated with a series of human-centric attributes.

*Work completed during internship at Qualcomm AI Research

†Qualcomm AI Research is an initiative of Qualcomm Technologies, Inc.

Algorithm 1: Neural style transformation network training

Inputs :

$\mathcal{D}$	a set of images
θ	initial parameters for encoder-decoder structure
E^c_θ, E^s_θ	a semantic encoder and a style encoder
MLP_θ, $G_{\theta,\gamma_1,\beta_1,...,\gamma_l,\beta_l}$	a statistics generator and a decoder w.r.t. AdaIN statistics
ϕ, D_ϕ	initial parameters for discriminator and the discriminator
K and N	a total number of optimization steps and a batch size

for $k = 1$ **to** K **do**
  $\mathcal{B} \leftarrow \{x_i \sim \mathcal{D}\}_{i=1}^N$
  for $x_i \in \mathcal{B}$ **do**
    $s_i \sim N(0, \mathbf{I_m})$ // sample random style map
    $c_{x_i} \leftarrow E^c_\theta(x_i)$ // encodes semantics
    $\gamma^s_1, \beta^s_1, ..., \gamma^s_l, \beta^s_l \leftarrow MLP(s_i)$ // compute statistics for s
    $x'_i \leftarrow G_{\theta,\gamma_1^{s_i},\beta_1^{s_i},...,\gamma_l^{s_i},\beta_l^{s_i}}(c_{x_i})$ // generate a view
    $L_{dis,i} = ||\log(1 - D_\phi(x_i))||_1 + ||\log(D_\phi(x'_i))||_1$
  end
  $\delta\phi \leftarrow \frac{1}{N}\sum_{i=1}^{N} \partial_\phi L_{dis,i}$ // compute the total loss gradient w.r.t. ϕ
  $\phi \leftarrow \text{optimizer}(\phi, \delta\phi, \eta_k)$ // update discriminator parameters
  for $x_i \in \mathcal{B}$ **do**
    $c_{x_i} \leftarrow E^c_\theta(x_i)$, $s_{x_i} \leftarrow E^s_\theta(x_i)$, and $s_i \sim N(0, \mathbf{I_m})$
    $\gamma_1^{s_{x_i}}, \beta_1^{s_{x_i}}, ..., \gamma_l^{s_{x_i}}, \beta_l^{s_{x_i}} \leftarrow MLP(s_{x_i})$ and $\gamma_1^{s_i}, \beta_1^{s_i}, ..., \gamma_l^{s_i}, \beta_l^{s_i} \leftarrow MLP(s_i)$
    $\hat{x_i} \leftarrow G_{\theta,\gamma_1^{s_{x_i}},\beta_1^{s_{x_i}},...,\gamma_l^{s_{x_i}},\beta_l^{s_{x_i}}}(c_{x_i})$ and $x'_i \leftarrow G_{\theta,\gamma_1^{s_i},\beta_1^{s_i},...,\gamma_l^{s_i},\beta_l^{s_i}}(c_{x_i})$
    $c'_i \leftarrow E^c_\theta(x'_i)$ and $s'_i \leftarrow E^s_\theta(x'_i)$
    $L_{img,i} \leftarrow ||x_i - \hat{x_i}||_1$, $L_{c,i} \leftarrow ||c_i - c'_i||_1$, and $L_{s,i} \leftarrow ||s_i - s'_i||_1$
    $L_{adv,i} \leftarrow ||\log(1 - D_\phi(x'_i))||_1$
    $L_{gen,i} \leftarrow \lambda_{img} L_{img,i} + \lambda_c L_{c,i} + \lambda_s L_{s,i} + \lambda_{adv} L_{adv,i}$
  end
  $\delta\theta \leftarrow \frac{1}{N}\Sigma_{i=1}^{N} \partial_\theta L_{gen,i}$ // compute the total loss gradient w.r.t. θ
  $\theta \leftarrow \text{optimizer}(\theta, \delta\theta, \eta_k)$ // update generator parameters
end

Algorithm 2: View generation for encoder training

Inputs :

x	input
E^c_θ, MLP, G	a semantic encoder, a statistics generator, and a decoder
$\mathcal{T}_{geo}$	a distribution of geometric transformation

$t_{geo} \sim \mathcal{T}_{geo}$ // sample a geometric transformation
$c_x \leftarrow E^c_\theta(x)$ and $s \sim N(0, \sigma\mathbf{I_m})$
$\gamma^s_1, \beta^s_1, ..., \gamma^s_l, \beta^s_l \rightarrow MLP_\theta(s)$
$x' \leftarrow G_{\theta,\gamma^s_1,\beta^s_1,...,\gamma^s_l,\beta^s_l}(E^c_\theta(x))$
$\epsilon \sim Uniform(-\epsilon_{max}, \epsilon_{max})$ // sample a degree of transformation
$x'' \leftarrow (1 - \epsilon) * x + \epsilon * x'$ // apply linear augmentation to input image
$x''' \leftarrow t_{geo}(x'')$ // apply the sampled geometric transformation
Output: x'''

3.0.5 MNIST

The Modified National Institute of Standards and Technology database (MNIST) dataset is a large-scale dataset con-

Algorithm 3: Encoder training

Inputs :

$\mathcal{D}$	a set of images
θ	pretrained parameters for encoder-decoder structure
E^c_θ, MLP, G	a semantic encoder, a statistics generator, and a decoder
$\mathcal{T}_{geo}$	a distribution of geometric transformation
ψ, l_ψ	a projection head and its initial parameters
L_{cont}	an objective function of a contrastive learning method
K and N	a total number of optimization steps and a batch size

for $k = 1$ **to** K **do**
  $\mathcal{B} \leftarrow \{(x_{i,1}, x_{i,2})|x_{i,1}, x_{i,2} \sim \mathcal{D}\}_{i=1}^{N}$
  for $(x_{i,1},\ x_{i,2}) \in \mathcal{B}$ **do**
    $t_{geo,i,1}, t_{geo,i,2} \sim \mathcal{T}_{geo}$ `// sample a geometric transformation`
    $c_{x_{i,1}} \leftarrow E^c_\theta(x_{i,1})$, $c_{x_{i,2}} \leftarrow E^c_\theta(x_{i,2})$, and $s_{i,1}, s_{i,2} \sim N(0, \sigma\mathbf{I_m})$
    $\gamma_1^{s_{i,1}}, \beta_1^{s_{i,1}}, ..., \gamma_l^{s_{i,1}}, \beta_l^{s_{i,1}} \rightarrow MLP_\theta(s_{i,1})$
    $\gamma_1^{s_{i,2}}, \beta_1^{s_{i,2}}, ..., \gamma_l^{s_{i,2}}, \beta_l^{s_{i,2}} \rightarrow MLP_\theta(s_{i,2})$
    $x'_{i,1} \leftarrow G_{\theta,\gamma_1^{s_{i,1}},\beta_1^{s_{i,1}},...,\gamma_l^{s_{i,1}},\beta_l^{s_{i,1}}}(E^c_\theta(x_{i,1}))$
    $x'_{i,2} \leftarrow G_{\theta,\gamma_1^{s_{i,2}},\beta_1^{s_{i,2}},...,\gamma_l^{s_{i,2}},\beta_l^{s_{i,2}}}(E^c_\theta(x_{i,2}))$
    $\epsilon_{i,1}, \epsilon_{i,2} \sim Uniform(-\epsilon_{max}, \epsilon_{max})$
    $x''_{i,1} \leftarrow (1-\epsilon) * x_{i,1} + \epsilon * x'_{i,1}$ and $x''_{i,2} \leftarrow (1-\epsilon) * x_{i,2} + \epsilon * x'_{i,2}$
    $x'''_{i,1} \leftarrow t_{geo}(x''_{i,1})$ and $x'''_{i,2} \leftarrow t_{geo}(x''_{i,2})$
    $L_{enc,i} \leftarrow L_{cont}(l_\psi(E_\psi(x'''_1), E_\psi(x'''_2)))$
  end
  $\delta\psi \leftarrow \frac{1}{N}\Sigma_{i=1}^{N}\partial_\psi L_{enc,i}$ `// compute the total loss gradient w.r.t.` ψ
  $\psi \leftarrow \text{optimizer}(\psi, \delta\psi, \eta_k)$ `// update generator parameters`
end

sists of 60,000 train images and 10,000 test images. The digits have been size-normalized and centered in a fixed-size image.

3.0.6 FaMNIST

Fashion-MNIST is a dataset consists of a training set of 60,000 examples and a test set of 10,000 examples. Each example is a 28×28 grayscale image, associated with a label from 10 classes.

3.0.7 CUBirds

Caltech-UCSD Birds-200-2011, which we call CUBirds in short, is a dataset widely-used for fine-grained visual categorization task. It consists of 11,788 images for 200 bird species, 5,994 for training and 5,794 for testing.

3.0.8 VGGFlower

The VGGFlower dataset is a flower datasets consisting of 102 different categories of flowers ommonly occuring in the United Kingdom. Each class consists of between 40 and 258 images.

3.0.9 TrafficSign

The German Traffic Sign Recognition Benchmark, which we call TrafficSign in short, consists of 43 classes of traffic signs splitting into 39,209 training images and 12,630 test images. The images have varying light conditions and rich backgrounds.

Algorithm 4: Transfer learning

Inputs :

$\mathcal{D}$	a set of images
$\dagger$	a label space
θ	pretrained parameters for encoder-decoder structure
E^c_θ, MLP_θ, G	a semantic encoder, a statistics generator, and a decoder
$\mathcal{T}_{geo}$	a distribution of geometric transformation
ξ, l_ξ	a task-specific prediction layer and its initial parameters
L_{trans}	an objective function of transfer learning
K and N	a total number of optimization steps and a batch size

for $k = 1$ **to** K **do**
  $\mathcal{B} \leftarrow \{(x_i, y_i)|x_i \sim \mathcal{D}, y_i \in \dagger\}_{i=1}^N$
  for $x_i \in \mathcal{B}$ **do**
    $t_{geo,i} \sim \mathcal{T}_{geo}$ `// sample a geometric transformation`
    $c_{x_i} \leftarrow E^c_\theta(x_i)$ and $s_i \sim N(0, \sigma\mathbf{I_m})$
    $\gamma_1^{s_i}, \beta_1^{s_i}, ..., \gamma_l^{s_i}, \beta_l^{s_i} \rightarrow MLP_\theta(s_i)$
    $x'_i \leftarrow G_{\theta,\gamma_1^{s_i},\beta_1^{s_i},...,\gamma_l^{s_i},\beta_l^{s_i}}(E^c_\theta(x_i))$
    $\epsilon_i \sim Uniform(-\epsilon_{max}, \epsilon_{max})$
    $x''_i \leftarrow (1-\epsilon) * x_i + \epsilon * x'_i$
    $x'''_i \leftarrow t_{geo}(x''_i)$
    $L_{trans,i} \leftarrow L_{trans}(l_\psi(E_\psi(x'''_i)), y_i)$
  end
  $\delta\xi \leftarrow \frac{1}{N}\Sigma_{i=1}^N \partial_\xi L_{trans,i}$ `// compute the total loss gradient w.r.t.` ξ
  $\xi \leftarrow$ optimizer$(\xi, \delta\xi, \eta_k)$ `// update generator parameters`
end

Table 1. Ablation study on geometric transformation configuration of our neural transformation framework. We compared encoder training for various view generation setup. 'perspective' denotes for a random perspective transformation. We randomly apply the rotation transformations 90-degree-wise. The performances are evaluated on the CIFAR-10 dataset.

resize&crop	horizontal flip	rotation	perspective	Encoder pretraining
				66.5
✓	✓			78.2
✓	✓	✓		52.8
✓	✓		✓	74.2