IEEE Open Journal of the Computer Society

Received 19 August 2024; accepted 10 September 2024. Date of publication 25 September 2024;
date of current version 25 October 2024. The review of this article was arranged by Associate Editor Song Guo.

Digital Object Identifier 10.1109/OJCS.2024.3462477

Photogenic Guided Image-to-Image Translation With Single Encoder

RINA OH AND T. GONSALVES (Member, IEEE)

Department of Information and Communication Sciences, Sophia University, Tokyo 102-8554, Japan

CORRESPONDING AUTHORS: RINA OH; T. GONSALVES (e-mail: rina_oh@sophia.ac.jp; t-gonsal@sophia.ac.jp).

ABSTRACT Image-to-image translation involves combining content and style from different images to generate new images. This technology is particularly valuable for exploring artistic aspects, such as how artists from different eras would depict scenes. Deep learning models are ideal for achieving these artistic styles. This study introduces an unpaired image-to-image translation architecture that extracts style features directly from input style images, without requiring a special encoder. Instead, the model uses a single encoder for the content image. To process the spatial features of the content image and the artistic features of the style image, a new normalization function called Direct Adaptive Instance Normalization with Pooling is developed. This function extracts style images more effectively, reducing the computational costs compared to existing guided image-to-image translation models. Additionally, we employed a Vision Transformer (ViT) in the Discriminator to analyze entire spatial features. The new architecture, named Single-Stream Image-to-Image Translation (SSIT), was tested on various tasks, including seasonal translation, weather-based environment transformation, and photo-to-art conversion. The proposed model successfully reflected the design information of the style images, particularly in translating photos to artworks, where it faithfully reproduced color characteristics. Moreover, the model consistently outperformed state-of-the-art translation models in each experiment, as confirmed by Fréchet Inception Distance (FID) and Kernel Inception Distance (KID) scores.

INDEX TERMS AI, deep learning, GAN, unpaired image-to-image translation, style synthesis translation.

I. INTRODUCTION

The image-to-image translation architectures using GAN (Generative Adversarial Networks) show promising results, such as in segmentation to real-world image translation [1], [2], one weather pattern into another weather pattern translation [3], [4], real-world photo to artistic image translation [2], [5], [6], super image resolution by concatenating the base of image and edge output [7], video super-resolution for satellite videos [8], and so on. The image-to-image translation deep learning techniques have developed thanks to the introduction of an unpaired image-to-image translation architecture such as CycleGAN [2]. This architecture allows the deep learning models to train translation without using a paired dataset.

Consider the image-to-image translation that synthesizes two inputs: the content image and the style image. In this article, this architecture is called the "Guided Image-to-Image Translation Architecture". The translated image provides visible clues to understand whether it has the spatial and style features akin to the input content and style images, respectively.

The guided image-to-image translation architecture also produces translation learning without a paired dataset. For instance, StarGAN v2 [9] and TSIT [10] employ encoders for the content image and the style image, and a single decoder for synthesizing the two encoded data. Almost all the existing guided image-to-image translation architectures use multiple encoders for the inputs. In this study, we attempted to implement a simpler architecture using only a single encoder for input content images.

The new proposed architecture uses a single pair of encoder and decoder in the Generator. In the encoder, our model encodes only the content image by multiple CNN layers; In the decoder, the whole style image is utilized through our original adaptive-instance normalization function called *DAdaINP: Direct Adaptive Instance Normalization with Pooling*. Since the model is inspired by TSIT and consists of a single encoder,

we named this guided image-to-image translation architecture *SSIT (Single-Stream Image-to-image Translation)*. We created not only the Generator, but also the Discriminator by adopting *ViT (Vision Transformer)* techniques whose input is the divided patches from the input image.

We trained the model for 3 different translation tasks: 1) Summer and Winter Translation: season appearance translation between summer and winter, 2) Time and Weather Translation in Driving: driving image translation based on the time and weather, and 3) Photo to Art Translation: real-world photos to artistic images translation.

Based on our results, we could investigate the following:

1) Even if only a single encoder is used for content images and the style images are proceeded with simple layers, the trained model could perform style synthesizing output following the design features of the input style image.
2) In each experiment in this study, our model could achieve a lower FID and KID scores than the other models in almost every target style.

The implementation of our model can be accessed from Github: https://github.com/rkomatsu2020/SSIT2023

II. RELATED WORKS

This section introduces the related works that translate images through the deep learning neural networks.

A. NEURAL STYLE TRANSFER

Convolution Neural Network (CNN) contributed to the first appearance of image-to-image translation. CNN is used to extract spatial features from the content image and texture features from the style image.

Gatys et al. [11], [12] proposed a method called Neural Style Transfer (NST). NST uses the deep convolution neural network VGG-19 [13], which consists of 16 CNNs and 5 max pooling layers. NST combines the spatial features of the content image from the deep layer of the VGG network and the texture features of the style image from each CNN layer.

Translation with NST is based on an optimization-based method which means repeated optimization image generation. Johnson et al. [14] point out that the optimization-based method requires a lot of time for the forward and backward passes through the pre-trained network. Shen et al. [15] proposed the smart file size of image translation architecture by using meta-network to obtain style image features and output translated images in a shorter time than the optimization-based method transfer.

These earlier methods could only handle a limited number of trained styles. NST methods were developed to handle the translation of multiple styles from a single input image. For example, Chen et al. [16] proposed StyleBank layer that complies multiple style features for each specific style in its translation architecture. Chen and Schmidt [17] introduced swapping the style method in which the middle features of the content image are replaced by the features of the style image patch-by-patch, allowing arbitrary translation.

B. IMAGE-TO-IMAGE TRANSLATION ARCHITECTURES BASED ON GAN

In creative tasks using deep learning, Generative Adversarial Network (GAN) is needed to build an image-to-image translation architecture.

GAN consists of a Generator network to generate images, and a Discriminator network to discriminate the inputs [18]. The Discriminator is trained to distinguish whether the inputs are from the image dataset (real images) or are being generated by the Generator (fake images). On the other hand, the Generator is trained to generate and output images to fool the Discriminator into recognizing them as real.

Related studies using GAN provides results that output realistic images via a trained Generator such as handwritten characters [19], medical images for retinal vessels [20], and anime style images [21], [22].

The most difficult task in adversarial learning is to strike a balance between the Generator and the Discriminator. If the equilibrium becomes unstable, something like mode collapse may occur, making the Generator output images that are not like real-world images, but rather distorted images. Several methods have been proposed to solve this phenomenon: Unrolled GAN provides the adversarial Generator with a more costly gradient descent by repeating the generation and discrimination step for the Generator [23]; WGAN-GP computes the gradient penalty for the Discriminator during the discrimination phase and enforces the Lipschitz constraint [24]. Further, Takeru Miyato et al. [25] proposed the normalization called Spectral Normalization, which tunes the hyper-parameters in the model and preserves the Lipschitz constant.

Our proposed model also refers to utilizing this Spectral Normalization in our Discriminator.

C. MULTI DOMAIN IMAGE-TO-IMAGE TRANSLATION ARCHITECTURES

To achieve outputting multiple style images using a single Generator, a conditional vector with one-hot presentation is also input to the Generator as another input. This method is the same as in Conditional GAN (cGAN) which trains the Generator to output conditional images by inputting random noise seed and conditional one-hot vector related to the targeted category [27].

StarGAN [28] trains the Generator to output the conditional images with the features of the target domain through adversarial loss and classification loss by the Discriminator.

D. GUIDED IMAGE-TO-IMAGE TRANSLATION ARCHITECTURES

Multi-domain image-to-image translation architectures tend to be "crested", because the Generator learns the "common" features from all the style images in each domain, and not the "noble" features of each style image. This production style synthesizing method is not only NST, but also the image-to-image translation architecture with GAN.

In the encoding phase, the encoder aims to output some special features from the input. StarGAN-v2 [9], MUNIT [31], DRIT [32], and DRIT++ [33] use double encoders to extract the style features (attribute features) and the spatial content features from a single input and synthesize the content features from the input content image and the style features from the target style image. StarGAN-v2 [9] and MUNIT [31] have a content encoder that convolutes the input through CNN and the style encoder which convolutes and outputs the flattened style features through MLP. On the other hand, DRIT [32] and DRIT++ [33] have the same structures in the content encoder and style encoder and obtain the channel-wise content feature maps and style feature maps.

In the decoding phase, the encoded content and style features are gathered through a process (such as normalization) in the decoder. Adaptive Instance Normalization (AdaIN) [34] which computes the affine parameters (shifting and scaling) from the style input, is the representative normalization function for decoding with encoded content and style features. Combining AdaIN in image-to-image translation architecture is utilized in U-GAT-IT [36].

Technical normalizations other than AdaIN are also in use. For example, TSIT [10] uses element-wise feature adaptive denormalization (FADE) using the previous content feature maps to preserve the content spatial structure and feature adaptive instance normalization (FAdaIN) to instruct style information. Cho et al. employed GDWCT (Group-wise deep whitening-and-coloring transformation) which conducts the translation through content features, the encoded style information matrix of coloring transformation and the mean in the bottom residual blocks between the encoder and the decoder [37].

However, to synthesize them, the Generators in these architectures almost always employ two encoders: one for the content images and the other for the style images. Therefore, with the intention of creating a simpler Generator, the authors tried implementing a single encoder whose input is only the content image and a single decoder whose conditional input is the entire style image.

III. PROPOSED MODEL: SSIT (: SINGLE-STREAM IMAGE-TO-IMAGE TRANSLATION)

Following TSIT [10], we implemented a guided image-to-image translation architecture. Unlike TSIT, our model does NOT have the same length layers for content encoder and style encoder. We only encode content images by multiple convolution layers. Therefore, we named our proposed model SSIT (Single-Stream Image-to-Image Translation). This section introduces the details of the architecture, the objective loss functions and the hyperparameters for conditional translation training.

A. THE ARCHITECTURES

Our SSIT consists of a Generator and a Discriminator. The Discriminator aims to discriminate whether the input is real (belongs to the style image dataset Y) or fake (generated by the Generator). The Generator in Fig. 1, on the other hand, aims to output the synthesized image $G(x, y)$ which has the content spatial features from the input content image $x \in X$, and the style design features from the style image $y \in Y$. Moreover, the Generator aims to fool the Discriminator in believing that $G(x, y)$ is real.

To enable synthesizing the style features to the bottom and decoder, we implemented the original normalization function named DAdaIN: Direct Adaptive Instance Normalization with Pooling. The features of the entire spatial of style image is extracted through different pooling layers, such as adaptive max pooling and adaptive average pooling. As pooling layers can contribute towards pointing to the characteristic region such as CAM (Class Activation Maps) [38], we suggest pooling layers have the potential to extract important design. Also, we employed the residual block with the skip connection in the bottom to avoid the degradation problem caused by the deep and complicated structure in the neural network [39]. Furthermore, to make our Generator capable of outputting various translated images, we added Gaussian Noise layer prior to each CNN layers referring to StyleGAN's techniques in [35], [40].

Fig. 2 is an overview of the Discriminator. We implemented a single discriminator with narrow Vision Transformer (ViT) architecture which consists of the multiple-head attention modules using embedded patches input [41]. ViT has transformer encoder whose input is the divided patches and the multi-head attention modules which consists of multiple self-attention calculation processes using the query, value and key. We employed this model expecting ViT to have the potential to consider the entire spatial features from content images and extract the characteristic features through the attention modules. Also, to train adversarial learning with stability, we adopt spectral normalization in first embedding linear layer and each multi-head attention layers different from related studies MUNIT [31], GDWCT [37], and TSIT [10]. As our model aims to make the construction simpler, we set a single discriminator and compute the adversarial loss from the single size of feature map D_{adv}.

B. DADAINP: DIRECT ADAPTIVE INSTANCE NORMALIZATION WITH POOLING

To establish our Generator with a single encoder for content images, we implemented the normalization which utilizes the entire style image. Fig. 3 shows the overview of our proposed normalization: Direct Adaptive Instance Normalization with Pooling.

As the base, we employed AdaIN (adaptive-instance normalization) for flexibility in style interpolation in the feed-forward neural network [32], [34].

C. OBJECTIVE LOSS FUNCTIONS

In the training phase, we adapted (1) for our Generator and (2) for our Discriminator.

$$L_G = \lambda_{adv} * L_{adv}(G) + \lambda_c L_c^{ViT} + \lambda_s * L_s^{ViT} \quad (1)$$

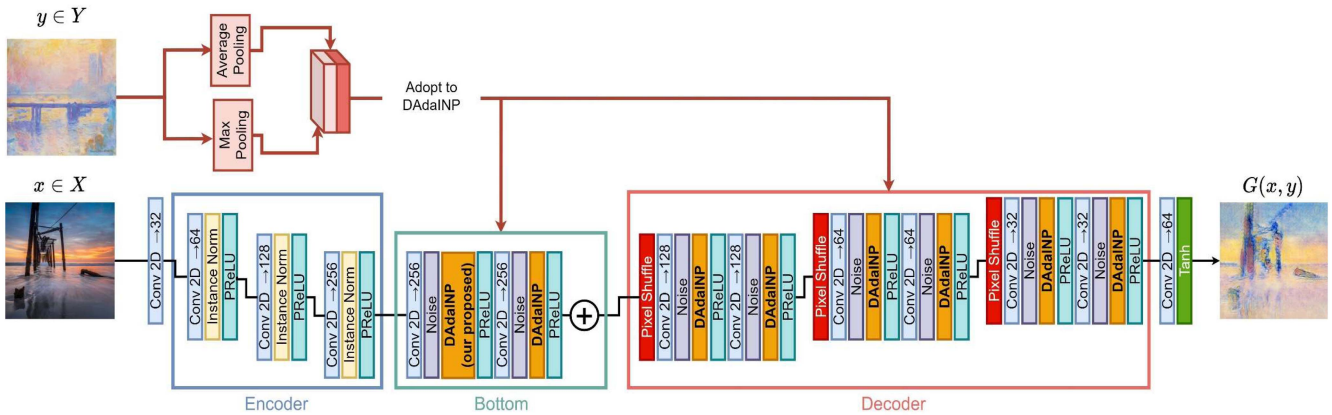


FIGURE 1. The overview of our SSIT's generator.

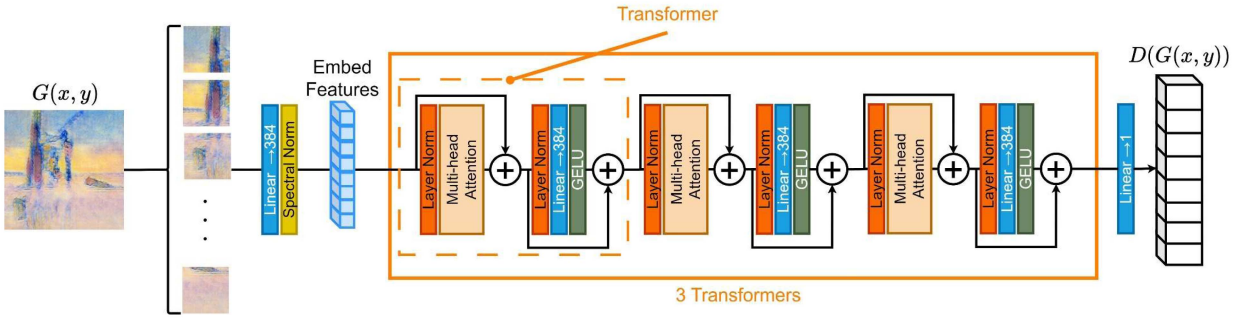


FIGURE 2. The overview of our SSIT's discriminator.

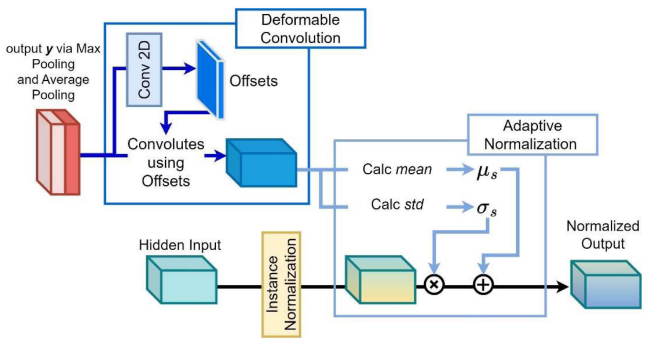


FIGURE 3. The overview of our direct adaptive instance normalization.

$$L_D = \lambda_{adv} * L_{adv}(D) \tag{2}$$

L_{adv} is the adversarial loss based on 2 contents: 1) LSGAN method whose criterion is the least square loss between the real and the fake images [42] and 2) One-sided label smoothing which sets 0 as fake label and 0.9 as real one [43] expecting the advantage of smoothing to prevent the model from being too confident about prediction and being over-fitted following in [44]. We employed this method for stable adversarial training. $L_{adv}(D)$ and $L_{adv}(G)$ compute the loss using the feature map through transformers. The details of the Discriminator's adversarial losses are shown in (3). The Generator's losses are shown in (4).

$$L_{adv}(D) = \mathbb{E}_{y \sim p_{data}(y)}\left[(0.9 - D(y))^2\right] + \mathbb{E}_{x \sim p_{data}(x), y \sim p_{data}(y)}\left[(D(G(x, y)))^2\right] \tag{3}$$

$$L_{adv}(G) = \mathbb{E}_{x \sim p_{data}(x), y \sim p_{data}(y)}\left[(0.9 - D(G(x, y)))^2\right] \tag{4}$$

In calculating the spatial content differences and the design style ones, we used the pretrained classification model DINO-ViT dividing the patch size in 16 × 16 [45]. We referred to [46] for utilizing pre-trained ViT to calculate losses. As [46] targeted single pair of content and style images, we tried expanding this method for Guided image-to-image translation.

L_c^{ViT} in (5) utilizes the output feature maps: $F_{map}(\ldots)$ excepting class token from 7th transformer encoder in pretrained ViT model. We compare the inner products between the similarity matrixes: translated images and content images. The similarity matrix is based on the inner product between $F_{map}(\ldots)$ and transposed one: $F_{map}(\ldots)^T$. L_s^{ViT} in (6) calculates the perceptual loss by comparing the class tokens comes from last transformer encoder by L2 norm.

$$L_c^{ViT} = \left\| sim\left(F_{map}(G(x, y)), F_{map}(G(x, y))^T\right) - sim\left(F_{map}(x), F_{map}(x)^T\right)\right\|_{L1} \tag{5}$$

$$L_s^{ViT} = \left\| F_{cls}(G(x, y)) - F_{cls}(y)\right\|_{L2} \tag{6}$$

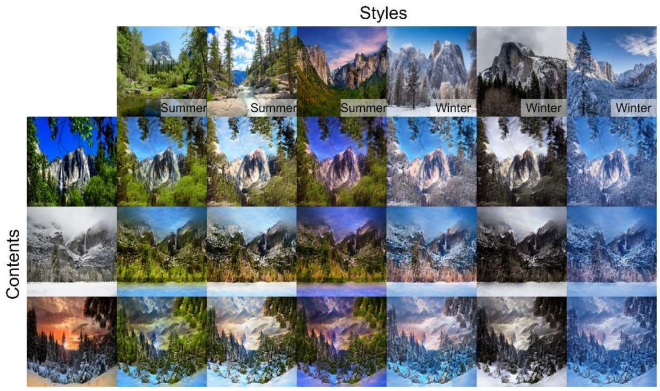


FIGURE 4. Visualized translating result by inputting content image and multiple style images using trained SSIT for summer and winter translation (leftmost images are content images and top images are input style images which belong to summer & winter domains).

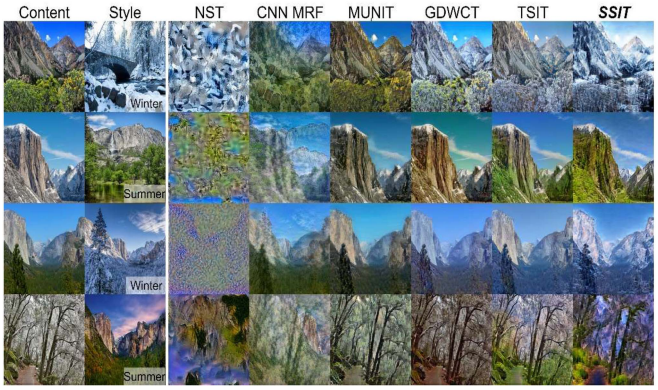


FIGURE 5. Comparing result by inputting content image and multiple style images among models in summer and winter translation.

D. HYPERPARAMETERS FOR TRAINING PHASE

As for the hyperparameters in translation training, we set the batch size = 15 in the experiments: "*Summer and Winter Translation*" and "*Photo-to-Art Translation*" and batch size = 8 in "*Time and Weather Translation in Driving*". The sizes we adapt in each experiment are introduced in Section IV *EXPERIMENTAL SETUP*.

As the optimization function for our Generator and Discriminator, we employed Adam [47] setting the learning rate in the Generator: $l_G = 0.0001$ and the Discriminator: $l_D = 0.0004$ respectively; and the betas $(\beta_1,\ \beta_2) = (0.0,\ 0.9)$ in both the cases.

IV. EXPERIMENTAL SETUP

In the synthesizing study using our proposed model, we prepared 3 kinds of image-to-image translations. This section explains the details of the content/style images that were used in each experiment.

A. DATASETS

1) SUMMER AND WINTER TRANSLATION

This translation experiment aims to translate the content images taken in summer (or winter) to likely winter (or summer) images by inputting the style images with the opposite season as the content.

We employed the landscape photos from Frickr as in the CycleGAN study [2]. Our model trained with 1231 summer images and 962 winter images. In all, we tested 309 summer and 238 winter images. In this experiment, we set the image size $(H,\ W) = (256\ px,\ 256\ px)$ and trained for 150 epochs. We set loss weights $\lambda_s = 0.1,\ \lambda_c = 3.0,\ \lambda_{adv} = 1.0$.

2) TIME AND WEATHER TRANSLATION IN DRIVING

This translation experiment aims to translate a single image taken from the front of a car in 6 different time and weather conditions: daytime & clear, daytime & rainy, daytime & snowy, nighttime & clear, nighttime & rainy, and nighttime & snowy.

We downloaded the BDD100K driving image dataset [48] and extracted the images using the labels based on weather and time information. In all, we were able to collect the training dataset: 12454 (daytime & clear), 2522 (daytime & rainy), 2862 (daytime & snowy), 22884 (nighttime & clear), 2208 (nighttime & rainy), and 2248 (night & snowy) and the test data set: 1764 (daytime & clear), 396 (daytime & rainy), 422 (daytime & snowy), 3724 (night & clear), 286 (night & rainy), and 273 (night & snowy).

In this experiment, we set the image size (H, W) = (256 px, 512 px) and trained for 30 epochs. We set loss weights $\lambda_s = 3.0,\ \lambda_c = 1.0,\ \lambda_{adv} = 1.0$.

3) PHOTO-TO-ART TRANSLATION

This translation aims to translate the single landscape photo into the painting styles of famous artists by entering the artist's artwork as a style image.

As for the content images, we used the landscape images from Flicker as shown by CycleGAN in the experiment "Photo ↔ Art for style transfer". From this dataset, we used 6287 content images for training and 751 images for translation evaluation. For the style images, the artworks of famous artists, we focused on 10 types of art styles: Pissarro, Monet, Matisse, Picasso, Cezanne, Gauguin, Renoir, Studio Ghibli, Ukiyo-e, and van Gogh. The 1128 artworks of Studio Ghibli were collected from the official website [49]. For the others, we collected the artworks of famous artists from the Kaggle competition site "Painters by Numbers" [50]. Using this dataset, we were able to extract the artworks based on the attached annotation file: 497 (Pissarro), 498 (Monet), 415 (Matisse), 431 (Picasso), 493 (Cezanne), 473 (Gauguin), 462 (Renoir), 350 (Ukiyo-e), and 374 (van Gogh). We set the image size $(H,\ W) = (256\ px,\ 256\ px)$ and trained for 150 epochs. We set loss weights $\lambda_s = 3.0,\ \lambda_c = 1.0,\ \lambda_{adv} = 1.0$.

FIGURE 6. Visualized translating result by inputting content image and multiple style images using trained SSIT for time and weather translation in driving.

FIGURE 7. Comparing result by inputting content image and multiple style images among models in time and weather translation in driving.

B. HARDWARE

To be able to compute the image processing as fast as possible, we employed *NVIDIA RTX A6000* GPU with a memory of 48 GB.

V. RESULTS

In this section, the results of applying our proposed model are presented in 2 parts.

First, the translated images are visualized by inputting the content image from the test dataset and the target image. Second, the evaluation results are used to measure the extent to which the translated images have the unique features of the target domain and how diverse they are when different style images are input to the system. We compared the results with 5 different models. (Combining content and style features through optimization: NST [11], [12] and CNNMRF [51] and image-to-image translation architectures based on GAN: MUNIT [31], GDWCT [37], and TSIT [10])

A. VISUALIZED RESULTS

1) SUMMER AND WINTER TRANSLATION

Fig. 4 shows the visualized results output by the trained SSIT for the summer and winter translation experiments. Our trained model could give reasonable results considering which image is added. For example, if the summer image with lush trees is added to the winter image in the second column of the content, the synthesized images will have the lush trees almost the same as the input images. Also, our model reflects the color of the sky that each style image has in the synthesized results. Fig. 5 shows the results of the comparison with other models.

Comparison between the models with optimization method and those with GAN show that the latter gives slightly more stable results, considering how many of the spatial content features remain. Our proposed model, SSIT was able to output the images reflecting the design features of each style image (e.g., the sky in style images) almost the same as GDWCT, even though our model has only a single encoder.

2) TIME AND WEATHER TRANSLATION IN DRIVING

Fig. 6 shows the visualized results output by our trained SSIT for time and weather translation in the driving test. The trained model could work especially for translation based on the time domain. For example, when the content image from the night domain is combined with the style image from the day and clear domains, the output image becomes bright and renders the blue sky in the upper domain like the input style image. Not only when translating the content image under the guided style image with the daytime & clear domain, but also when entering the style image with the daytime & rainy domain,

FIGURE 8. Visualized translating result by inputting content image and multiple style images using trained SSIT for photo-to-art translation.

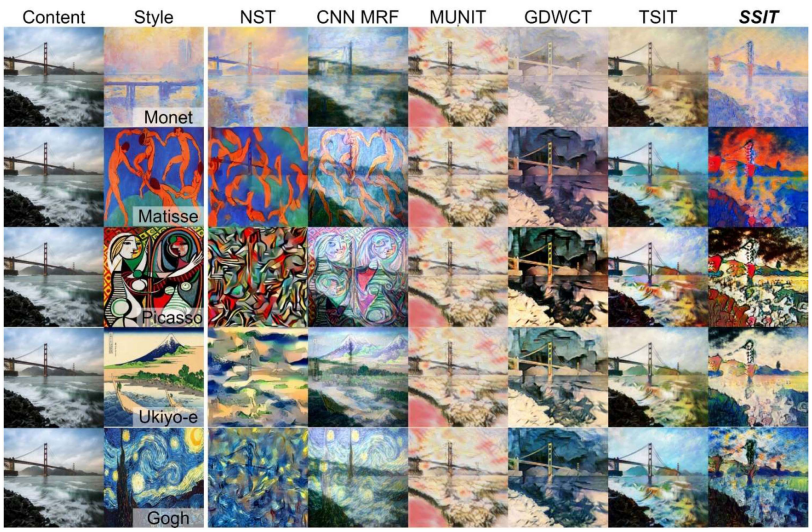


FIGURE 9. Comparing result by inputting content image and multiple style images among models in photo-to-art translation.

the translated image renders a slightly bright landscape and cloudy sky like the style image. However, when translating the content image into the nighttime & rainy and nighttime & snow domains, the weather features, such as wet ground due to rain and piled-up snow in the dark, are not well described in the translated image.

Fig. 7 shows the results of the comparison between the models. The results show that the image-to-image translation models using GAN can almost completely reproduce the sky coloration of a daylight style image. However, they seemed to have difficulty translating content images into night & rain and night & snow styles. This is probably because the discriminator has become too strict in distinguishing the ground if it has a strange texture.

3) PHOTO-TO-ART TRANSLATION

Fig. 8 shows the visualized results output by our SSIT experiment for translating photographs into art. Each visualized image exhibits the spatial features of the content image, such as the shape of the objects. In addition, each output image exhibits unique painting characteristics resulting from the color scheme and edge strength of the input artwork (style image). For example, when translating the content images into the artwork "View of Collioure (1905)" by Mattisse, each output image might have a pale pink coloration. Similarly, when translating to Picasso's artwork "Seated Woman with Blue and Red Hat (1939)", the output images could describe a strong edge separating the light side from the shaded side. The proposed Direct Adaptive Instance Normalization with Pooling function could work when considering the characteristic design of the input image through the convolution process.

Fig. 9 shows the comparison between the models (the resulting image is placed in Appendix due to its large size). Looking at the results of each model, NST tends to depend on the content loss and style loss hyperparameters to obtain a moderately translated image that has some spatial content features (we set fixed values for the hyperparameters in NST). On the other hand, CNNMRF can preserve the spatial content features more than NST. But the translation approach of

TABLE 1. Comparing FID, KID (KID With Mean(x100) ± Std) and LPIPS Scores Among Models in Summer and Winter Translation

	FID Score ↓ score is better				KID Score (Mean(x100) ± Std) ↓ score is better				LPIPS Score ↑ score is better			
	MUNIT	GDWCT	TSIT	SSIT (ours)	MUNIT	GDWCT	TSIT	SSIT (ours)	MUNIT	GDWCT	TSIT	SSIT (ours)
Summer	251.98	319.12	111.98	**89.14**	25.23±0.03	33.19±0.02	4.51±0.01	**3.44**±0.01	0.4702	**0.748**	0.3629	0.4723
Winter	217.63	307.82	100.64	**97.7**	18.93±0.02	30.3±0.02	**2.92**±0.01	3.62±0.01	0.4748	**0.7567**	0.3626	0.4548

TABLE 2. Comparing FID, KID (KID With Mean(x100) ± Std) and LPIPS Scores Among Models in Summer and Winter Translation

	FID Score ↓ score is better				KID Score (Mean(x100) ± Std) ↓ score is better				LPIPS Score ↑ score is better			
	MUNIT	GDWCT	TSIT	SSIT (ours)	MUNIT	GDWCT	TSIT	SSIT (ours)	MUNIT	GDWCT	TSIT	SSIT (ours)
Daytime & Clear	210.13	257.91	**112.04**	142.02	24.8±0.02	30.08±0.02	**11.98**±0.02	17.03±0.03	0.4709	**0.5916**	0.365	0.2604
Daytime & Rainy	194.53	246.84	**110.56**	116.39	17.35±0.02	23.56±0.02	**7.14**±0.01	9.15±0.02	0.4944	**0.6031**	0.3539	0.2873
Daytime & Snowy	207.39	258.34	**119.9**	125.91	19.1±0.02	24.82±0.02	**8.33**±0.01	10.33±0.02	0.4963	**0.6026**	0.3557	0.2871
Night & Clear	221.52	232.8	137.56	**76.39**	28.44±0.03	27.83±0.02	16.61±0.03	**7.95**±0.02	0.5611	**0.6388**	0.3286	0.3635
Night & Rainy	181.01	246.53	111.52	**70.38**	15.62±0.03	23.31±0.02	6.91±0.02	**2.13**±0.01	0.5018	**0.6063**	0.3526	0.2959
Night & Snowy	190.15	259.47	112.91	**63.99**	17.72±0.02	25.35±0.02	7.36±0.01	**1.69**±0.01	0.502	**0.6065**	0.3527	0.2961

TABLE 3. Comparing FID, KID (KID With Mean (x100) ± Std) and LPIPS Scores Among Models in Photo-to-Art Translation

	FID Score ↓ score is better				KID Score (Mean(x100) ± Std) ↓ score is better				LPIPS Score ↑ score is better			
	MUNIT	GDWCT	TSIT	SSIT (ours)	MUNIT	GDWCT	TSIT	SSIT (ours)	MUNIT	GDWCT	TSIT	SSIT (ours)
Pissarro	226.71	173.64	**138.57**	142.01	18.04±0.02	10.85±0.01	**6.34**±0.01	7.62±0.01	**0.673**	0.6526	0.4412	0.6082
Monet	198.07	150.02	**111.02**	124.13	16.03±0.02	9.19±0.01	**3.76**±0.01	5.97±0.01	**0.673**	0.6526	0.4411	0.6085
Matisse	245.76	219.45	213.56	**211.07**	19.35±0.02	14.87±0.02	**12.41**±0.02	13.64±0.01	**0.673**	0.6526	0.4411	0.6086
Picasso	256.35	220.69	223.56	**210.56**	21±0.02	15.3±0.02	13.43±0.01	**13.38**±0.01	**0.673**	0.6526	0.4411	0.6087
Cezanne	225.44	153.2	169.8	**150.89**	17.58±0.02	8.09±0.01	8.62±0.01	**7.81**±0.01	**0.673**	0.6526	0.4412	0.6086
Gauguin	213.14	164.24	162.77	**146.36**	15.94±0.02	9.73±0.02	8.81±0.02	**7.6**±0.01	**0.673**	0.6526	0.4412	0.6086
Renoir	218.9	187.64	**160.93**	167.4	15.62±0.02	10.92±0.02	**6.73**±0.01	8.9±0.02	**0.673**	0.6526	0.4412	0.6084
Ghibli	252.13	250.48	**191.19**	229.67	22.38±0.02	20.4±0.02	**11.76**±0.02	17.82±0.01	**0.673**	0.6526	0.4412	0.6082
Gogh	230.82	185.94	177.63	**175.33**	16.63±0.02	10.08±0.01	**7.47**±0.01	8.68±0.01	**0.673**	0.6526	0.4412	0.6084
Ukiyo-e	235.31	222.91	**204.65**	210.24	22.84±0.02	18.95±0.02	**15.36**±0.01	16.59±0.01	**0.673**	0.6526	0.441	0.6084

CNNMRF only seemed to superimpose the style image layer over the content images without changing the coloring of the objects in the content images, especially when considering the synthesized results with the artworks of Matisse and Picasso. The optimization method showed some of the style features in the output images.

B. SCORE EVALUATIONS

The mere visualization of the synthesized images is not sufficient to prove that our SSIT is able to generate synthesized images in the probabilistic style from the input content.

In this section, we introduce the score values to estimate 2 points: 1) How close are the conditional synthesized images to the targeted style images in terms of characteristic features, and 2) What is the diversity of synthesized images from a single input content image by adding multiple style images to the generator. To evaluate them, we used KID (Kernel Inception Distance) for 1) and LPIPS (Learned Perceptual Image Patch Similarity) for 2).

To compute these scores, we used the Python library "TorchMetrics" [52]. Since the NST and CNNMRF optimization methods take a long time to output the synthetic images,

TABLE 4. Comparing Computational Training Cost (MB) and FLOPs (x 10^9) Among Translation Experiences

		Computational Training Cost (MB)				FLOPs (x 10^9)			
		GDWCT (2 *Gs* and *Ds*)	MUNIT (2 *Gs* and *Ds*)	TSIT (1 *G* and *D*)	SSIT (ours) (1 *G* and *D*)	GDWCT (2 *Gs* and *Ds*)	MUNIT (2 *Gs* and *Ds*)	TSIT (1 *G* and *D*)	SSIT (ours) (1 *G* and *D*)
Summer to Winter Translation	net G	413.34*2	359.84*2	1347.67	**333.73**	27.96*2	66.33*2	42.37	**5.65**
	net D	42.58*2	42.58*2	42.57	**29.33**	4.17*2	2.08*2	2.08	**0.64**
Time and Weather Translation	net G	793.98*2	681.75*2	2016.34	**659.74**	55.89*2	132.66*2	84.74	**11.29**
	net D	63.2*2	63.2*2	63.02	**48.60**	2.08*2	4.17*2	4.16	**1.29**
Photo-to-Art Translation	net G	413.34*2	359.84*2	1347.67	**333.73**	27.96*2	66.33*2	42.37	**5.65**
	netD	42.93*2	42.93*2	42.57	**29.33**	2.09*2	2.09*2	2.08	**0.64**

in this section we compared the results of the image-to-image translation models: MUNIT, GDWCT, TSIT, and SSIT.

1) FID AND KID SCORES

FID and KID are the criteria to evaluate the characteristic feature similarity between the dataset of real images and the dataset of fake images (generated by Generator).

FID (Fréchet Inception Distance) [53] calculates the mean difference and covariance matrix between the extracted features of real and fake images. KID [54] calculates the maximum mean discrepancy between the extracted features of real images and fake images. KID can output a stable mean value when a small sample size of the image dataset is used.

2) LPIPS SCORE

LPIPS is the criterion to evaluate the spatial similarity between 2 patches based on perceptual loss. According to [55], the process of computing distance consists of extracting features of each patch from the trained model in ImageNet and finding the average value by multiplying the spatial L2 norm average by the pair patch. When the LPIPS score is close to 0, it means that the two patches are almost identical. Two inputs are used to calculate each score: Content images that DO NOT belong to the target style domain, and the style images. (For example, when testing the summer domain in summer and winter translation, we tested the winter images without including the summer images)

3) SCORE RESULTS

In result Tables 1 to 4, the best result score is written with red and bold text and the second score is written with blue and bold.

a) Summer and winter translation: Table 1 shows the FID, KID and LPIPS scores in Sumer and Winter Translation experiment. The results show that our proposed model could obtain the best score in the targeted styles. In addition, the visualization results demonstrate that our trained Generator could reflect the characteristics features like sky coloring, forest's greenery and the ground strongly. Thus, the visual results are also better than the previous studies.

Viewing LPIPS scores among various models, even if the output synthesized results using MUNIT have a few images different from the content images, the LPIPS score has become better than our SSIT. We suggest that the trained model in ImageNet for outputting and comparing both patches focused on the differences in the detailed features like slight deviation in pixels, rather than the entire features.

b) Time and weather translation in driving: Table 2 is the FID, KID and LPIPS scores in Time and Weather Translation in Driving experiment. In these score results, our model could mark the better results than the others too. Especially, considering the translation quality in time, our model could follow the sky coloring when translating daytime to night and vice versa. However, FID and KID scores are evaluated from the entire spatial features, and not detail and small features such as stacked snow in road. We need to consider architectures other than GAN.

LPIPS scores among the models also show that our model could not score higher than the others, even though the synthesized images differed from the content images. It scored less than 0.1. The reason is that the trained scoring model for LPIPS did not consider the whole spatial features of the image well.

c) Photo-to-art translation: Table 3 is the FID, KID and LPIPS scores in Photo-to-Art Translation experiment. Our model could obtain the best and the second-best scores in FID and KID score results. These results are achieved because the synthesized imaged by training our Generator could reflect the unique design features the input guided artworks considered from visualized results.

Although the LPIPS scores of our model were lower, the results show that the synthesized images could at least keep the content spatial features.

C. COMPUTATIONAL COST

To estimate the computational cost (based on the total number of calculation steps in forward process) for translation reduced by employing our Generator and Discriminator, we evaluated the models setting batch size = 1. Table 4 is the observed result table in comparing computational cost using the python library "torchinfo" [56]. Also, we calculated the FLOPS among models using the library "thop" [57].

Viewing the results, our Generator could reduce the computational cost and FLOPs than the other models. This

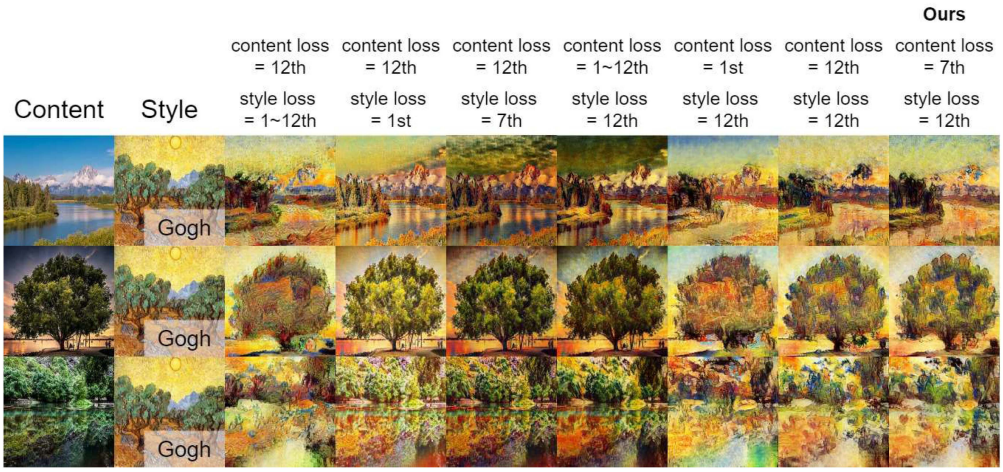


FIGURE 10. Comparing SSITs by changing extracted features from the transformer block in pre-trained DINO-ViT.

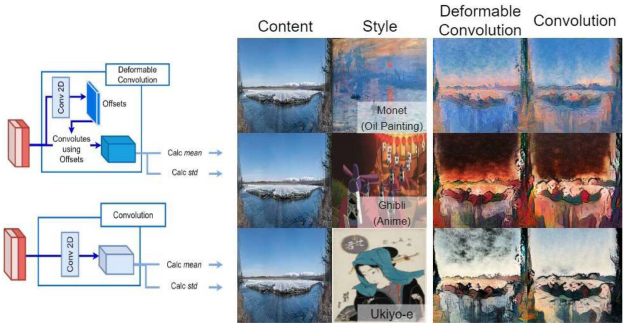


FIGURE 11. Comparing the translation performance in the structure of normalization in our decoder by feeding different artistic style inputs.

FIGURE 12. Tried Translating animal face to the other animal face using our trained SSIT.

achievement is because we employed a single encoder which consists of multiple convolution layers only for content image and placed pooling layers for extracting style features in different angles, not convolution layers. On the other hand, our discriminator tends to consume FLOPs. This is because the classification layer employs linear translation to output the classification result. Linear translation tends to consume more memory than convolution in calculation.

VI. DISCUSSION

A. DISCUSSION 1: HOW COULD OUR MODEL HOLD SPATIAL CONTENT FEATURES AND DESIGN STYLE FEATURES?

To calculate spatial content and design style losses, we extracted output features from the transformer blocks in a pre-trained DINO-ViT model. Our experiments showed that the trained SSIT could effectively synthesize both content and style features.

We observed how the pre-trained DINO-ViT reflects content and style features by extracting from different layers of transformer blocks (the DINO-ViT in [45] has 12 transformer blocks). Fig. 10 presents the comparison results. Extraction from shallower blocks tends to capture fewer style features, like edges and textures, while deeper blocks reflect more significant style features. Synthesized images using SSIT, which extract style loss from earlier blocks and content loss from deeper blocks, retain strong spatial content features.

For content loss, deeper block extraction preserves overall spatial content, allowing characteristic style designs to emerge. However, calculating style loss using all transformer blocks and content loss from only the last block results in images with weaker content features but stronger style features. The ViT's multi-head attention, processing image patches through inner products, captures global context features as it progresses to deeper layers. This suggests that integrating a pre-trained ViT in Guided Image-to-Image Translation effectively captures the overall structure of content and style images.

B. DISCUSS2: DEFORMABLE CONVOLUTION VS CONVOLUTION

Our model, DAdaINP, uses Deformable Convolution to calculate affine parameters instead of traditional Convolution. While standard Convolution uses fixed-sized receptive fields, Deformable Convolution adapts its receptive fields based on each input image. We compared the performance of DAdaINP

using Deformable Convolution with a version that uses standard Convolution.

Fig. 11 shows the comparison results. When guiding Monet and Ukiyo-e styles, the DAdaINP model with standard Convolution exhibited more uniform changes between light and shadow. In contrast, the model with Deformable Convolution produced more nuanced contrasts and enhanced detail, particularly at the horizon. For Anime style, Deformable Convolution achieved smoother coloring along background edges compared to standard Convolution.

Therefore, Deformable Convolution contributes to producing varied light and shadow contrasts and better reflects detailed features of the input style.

VII. CONCLUSION

This study introduced a new guided image-to-image translation model with a single encoder in the generator using our SSIT, demonstrating quality translation across various domains.

A. CHALLENGES1: FURTHER REDUCING THE CONSTRUCTION IN DISCRIMINATOR

We explored the use of ViT as a Discriminator to effectively assess whether synthesized images possess realistic features. Our proposed Generator and Discriminator showed strong performance in both visual results and scoring metrics. While we used three transformers in the Discriminator to minimize computational costs, this approach is more computationally intensive than using CNN layers. Deeper transformer blocks in ViT capture significant attention features, suggesting that additional transformers could further enhance translation quality. To balance quality improvement and computational cost, *CMT (Convolutional Neural Networks Meet Vision Transformers)* [58], which reduces computational overhead while maintaining performance, could be considered.

B. CHALLENGES2: EXTREME APPEARANCE TRANSLATION

Our model effectively translated landscape images into artistic ones and avoided "checkerboard artifacts" by using zero-order hold kernels [59] and sub-pixel convolution layers (pixel shuffle) [60]. However, when translating animal faces using the Animal Faces-HQ dataset [9], the model mainly altered coloring without fully capturing specific features like ear shapes. This limitation suggests that our model struggles with extreme appearance translations in localized areas (Fig. 12).

Given the current success of diffusion models, which generate high-quality images through denoising processes [61], [62], integrating these methods could provide more stable and flexible translations. Diffusion models, by progressively reconstructing images from noisy data, offer a more controlled generation process compared to GANs. To improve the ability to handle varied and complex translations under stable learning conditions, future work should explore the incorporation of diffusion models.

REFERENCES

[1] P. Isola, J.-Y. Zhi, T. Zhou, and A. A. Efros, "Image-to-image translation with conditional adversarial networks," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2017, pp. 1125–1134.

[2] J.-Y. Zhu, T. Park, P. Isola, and A. A. Efros, "Unpaired image-to-image translation using cycle-consistent adversarial networks," in *Proc. IEEE Int. Conf. Comput. Vis.*, 2017, pp. 2223–2232.

[3] X. Li, K. Kou, and B. Zhao, "WeatherGAN: Multi-domain weather translation using generative adversarial networks," 2021, *arXiv:2103.05422*.

[4] S. Hwang, S. Jeon, Y.-S. Ma, and H. Byun, "WeatherGAN: Unsupervised multi-weather image-to-image translation via single content-preserving UResNet generator," *Multimedia Tools Appl.*, vol. 81, pp. 40269–40288, 2022.

[5] W. Cho, S. Choi, D. K. Park, I. Shin, and J. Choo, "Image-to-image translation via group-wise deep whitening-and-coloring translation," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2019, pp. 10639–10647.

[6] R. Komatsu and T. Gonsalves, "Translation of real-world photographs into artistic images via conditional CycleGAN and StarGAN," *SN Comput. Sci.*, vol. 2, no. 6, pp. 1–20, 2021.

[7] K. Jiang, Z. Wang, P. Yi, G. Wang, T. Lu, and J. Jiang, "Edge-enhanced GAN for remote sensing image superresolution," *IEEE Trans. Geosci. Remote Sens.*, vol. 57, no. 8, pp. 5799–5812, Aug. 2019.

[8] Y. Xiao, X. Su, Q. Yuan, D. Liu, H. Shen, and L. Zhang, "Satellite video super-resolution via multiscale deformable convolution alignment and temporal grouping projection," *IEEE Trans. Geosci. Remote Sens.*, vol. 60, 2022, Art. no. 5610819.

[9] Y. Choi, Y. Uh, J. Yoo, and J.-W. Ha, "StarGAN v2: Diverse image synthesis for multiple domains," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2020, pp. 8188–8197.

[10] L. Jiang, C. Zhang, M. Huang, C. Liu, J. Shi, and C. C. Loy, "TSIT: A simple and versatile framework for image-to-image translation," in *Proc. Eur. Conf. Comput. Vis.*, 2020, pp. 206–222.

[11] L. A. Gatys, A. S. Ecker, and M. Bethge, "A neural algorithm of artistic style," *J. Vis.*, vol. 16, no. 12, p. 326, 2016, doi: 10.1167/16.12.326.

[12] L. A. Gatys, A. S. Ecker, and M. Bethge, "Image style transfer using convolutional neural networks," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2016, pp. 2414–2423.

[13] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," in *Proc. Int. Conf. Learn. Representations*, 2014, pp. 1–14.

[14] J. Johnson, A. Alahi, and L. F.-Fei, "Perceptual losses for real-time style transfer and super-resolution," in *Proc. Eur. Conf. Comput. Vis.*, 2016, pp. 694–711.

[15] F. Shen, S. Yan, and G. Zeng, "Neural style transfer via meta networks," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2018, pp. 8061–8069.

[16] D. Chen, L. Yuan, J. Liao, N. Yu, and G. Hua, "StyleBank: An explicit representation for neural image style transfer," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2017, pp. 1897–1906.

[17] T. Q. Chen and M. Schmidt, "Fast patch-based style transfer of arbitrary style," in *Proc. 30th Conf. Neural Inf. Process. Syst.*, 2016, pp. 1–5.

[18] I. J. Goodfellow et al., "Generative adversarial nets," 2014, *arXiv:1406.2661*.

[19] R. Komatsu and T. Gonsalves, "Conditional DCGAN's challenge: Generating handwritten character digit, alphabet and katakana," in *Proc. Annu. Conf. JSAI 33rd*, 2019, Art. no. 3B3E204.

[20] T. Iqbal and H. Ali, "Generative adversarial network for medical images (MI-GAN)," *J. Med. Syst.*, vol. 42, no. 11, pp. 1–11, 2018.

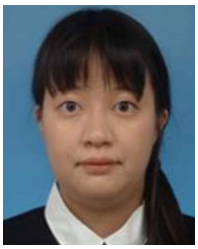
[21] B. Li, Y. Zhu, Y. Wang, C.-W. Lin, B. Ghanem, and L. Shen, "AniGAN: Style-guided generative adversarial networks for unsupervised anime face generation," *IEEE Trans. Multimedia*, vol. 24, pp. 4077–4091, 2022.

[22] C. Yang, L. Yu-Kun, and L. Yong-Jin, "CartoonGAN: Generative adversarial networks for photo cartoonization," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2018, pp. 9465–9474.

[23] L. Metz, B. Poole, D. Pfau, and J. Sohl-Dickstein, "Unrolled generative adversarial network," in *Proc. 5th Int. Conf. Learn. Representations*, 2017, pp. 1–25.

[24] I. Gulrajani, F. Ahmed, M. Arjovsky, V. Dumoulin, and A. Courville, "Improved training of Wasserstein GANs," in Proc. *Adv. Neural Inf. Process. Syst.*, 2017, vol. 30, pp. 1–20.

[25] T. Miyato, T. Kataoka, M. Koyama, and Y. Yoshida, "Spectral normalization for generative adversarial networks," in *Proc. 6th Int. Conf. Learn. Representations*, 2018, pp. 1–26.
[26] M. Sangwoo, C. Minsu, and S. Jinwoo, "InstaGAN: Instance-aware image-to-image translation," in *Proc. Int. Conf. Learn. Representations*, 2019, pp. 1–26.
[27] M. Mirza and S. Osindero, "Conditional generative adversarial nets," 2014, *arXiv:1411.1784*.
[28] Y. Choi, M. Choi, M. Kim, J.-W. Ha, S. Kim, and J. Choo, "StarGAN: Unified generative adversarial networks for multi-domain image-to-image translation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2018, pp. 8789–8797.
[29] P.-W. Wu, Y.-J. Lin, C.-H. Chang, E. Y. Chang, and S.-W. Liao, "RelGAN: Multi-domain image-to-image translation via relative attributes," in *Proc. IEEE/CVF Int. Conf. Comput. Vis.*, 2019, pp. 5914–5922.
[30] X. Yu, X. Chai, Z. Ying, T. Li, and G. Li, "SingleGAN: Image-to-image translation by a single-generator network using multiple generative adversarial learning," in *Proc. 14th Asian Conf. Comput. Vis.*, 2019. pp. 341–356.
[31] X. Huang, M.-Y. Liu, S. Belongie, and J. Kautz, "Multimodal unsupervised image-to-image translation," in *Proc. Eur. Conf. Comput. Vis.*, 2018, pp. 172–189.
[32] H.-Y. Lee, H.-Y. Tseng, J.-B. Huang, M. Singh, and M.-H. Yang, "Diverse image-to-image translation via disentangled representations," in *Proc. Eur. Conf. Comput. Vis.*, 2018, pp. 35–51.
[33] H.-Y. Lee et al., "DRIT++: Diverse image-to-image translation via disentangled representation," *Int. J. Comput. Vis.*, vol. 128, pp. 2402–2417, 2020.
[34] X. Huang and S. Belongie, "Arbitrary style transfer in real-time with adaptive instance normalization," in *Proc. IEEE Int. Conf. Comput. Vis.*, 2017, pp. 1501–1510.
[35] T. Karras, S. Laine, and T. Aila, "A style-based generator architecture for generative adversarial networks," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2019, pp. 4401–4410.
[36] K. Junho, K. Minjae, K. Hyeonwoo, and L. Kwanghee, "U-GAT-IT: Unsupervised generative attentional networks with adaptive layer-instance normalization for image-to-image translation," in *Proc. Int. Conf. Learn. Representations*, 2020, pp. 1–19.
[37] W. Cho, S. Choi, D. K. Park, I. Shin, and J. Choo, "Image-to-image translation via group-wise deep whitening-and-coloring transformation," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2019, pp. 10639–10647.
[38] B. Zhou, A. Khosla, A. Lapedriza, A. Oliva, and A. Torralba, "Learning deep features for discriminative localization," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2016, pp. 2921–2929.
[39] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2016, pp. 770–778.
[40] T. Karras, S. Laine, M. Aittala, J. Hellsten, J. Lehtinen, and T. Aila, "Analyzing and improving the image quality of StyleGANm," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2020, pp. 8110–8119.
[41] A. Dosovitskiy et al., "An image is worth 16x16 words: Transformers for image recognition at scale," in *Proc. 9th Int. Conf. Learn. Representations*, 2021, pp. 1–22.
[42] X. Mao, Q. Li, H. Xie, R. Y. K. Lau, Z. Wang, and S. P. Smolley, "Least squares generative adversarial networks," in *Proc. IEEE Int. Conf. Comput. Vis.*, 2017, pp. 2794–2802.
[43] T. Salimans, I. Goodfellow, W. Zaremba, V. Cheung, A. Radford, and X. Chen, "Improved techniques for training GANs," in Proc. *Adv. Neural Inf. Process. Syst.*, 2016, vol. 29, pp. 1–9.
[44] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, and Z. Wojna, "Rethinking the inception architecture for computer vision," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2016, pp. 2818–2826.
[45] M. Caron et al., "Emerging properties in self-supervised vision transformers," in *Proc. IEEE/CVF Int. Conf. Comput. Vis.*, 2021, pp. 9560–9660.
[46] N. Tumanyan, O. Bar-Tal, S. Bagon, and T. Dekel, "Splicing ViT features for semantic appearance transfer," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2022, pp. 10748–10757.
[47] D. P. Kingma and J. L. Ba, "Adam: A method for stochastic optimization," in *Proc. Int. Conf. Learn. Representations*, 2015, pp. 1–15.
[48] F. Yu et al., "BDD100K: A diverse driving dataset for heterogeneous multitask learning," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2020, pp. 2636–2645.
[49] Studio Ghibli, "新しくヽスタジオジブリ 5 作品の場面写真を追加提供致します - スタジオジブリ | STUDIO GHIBLI." 2020. [Online]. Available: https://www.ghibli.jp/info/013409/
[50] Small Yellow Duck and K. Nichol, "Painter by numbers." 2016. [Online]. Available: https://www.kaggle.com/c/painter-by-numbers
[51] C. Li and M. Wand, "Combining markov random fields and convolutional neural networks for image synthesis," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2016, pp. 2479–2486.
[52] N. S. Detlefsen et al., "TorchMetrics - measuring reproducibility in PyTorch," *J. Open Source Softw.*, vol. 7, no. 70, 2022, Art. no. 4101.
[53] M. Bińkowski, D. J. Sutherland, M. Arbel, and A. Gretton, "Demystifying MMD GANs," in *Proc. Int. Conf. Learn. Representations*, 2018, pp. 1–36.
[54] M. Heusel, H. Ramsauer, T. Unterthiner, and B. Nessler, "GANs trained by a two time-scale update rule converge to a local nash equilibrium," in *Proc. Adv. Neural Inf. Process. Syst.*, 2017, vol. 30, pp. 1–12.
[55] R. Zhang, P. Isola, A. A. Efros, E. Shechtman, and O. Wang, "The unreasonable effectiveness of deep features as a perceptual metric," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2018, pp. 586–595.
[56] GitHub, "torchinfo." 2022. [Online]. Available: https://github.com/TylerYep/torchinfo
[57] Github, "thop." 2018. [Online]. Available: https://github.com/Lyken17/pytorch-OpCounter
[58] J. Guo et al., "CMT: Convolutional neural networks meet vision transformers," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2022, pp. 12175–12185.
[59] Y. Sugawara, S. Shiota, and H. Kiya, "Checkerboard artifacts free convolutional neural network," *APSIPA Trans. Signal Inf. Process.*, vol. 8, 2019, Art. no. e9.
[60] W. Shi et al., "Real-time single image and video super-resolution using an efficient sub-pixel convolutional neural network," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2016, pp. 1874–1883.
[61] J. S. Dickstein, E. A. Weiss, N. Maheswaranathan, and S. Ganguli, "Deep unsupervised learning using nonequilibrium thermodynamics," in *Proc. Int. Conf. Mach. Learn.*, 2015, pp. 2256–2265.
[62] J. Ho, A. Jain, and P. Abbeel, "Denoising diffusion probabilistic models," in Proc. *Adv. Neural Inf. Process. Syst.*, 2020, vol. 33, pp. 6840–6851.



RINA (KOMATSU) OH received the Ph.D. degree from Sophia University, Tokyo, Japan, by developing and proposing a novel N-to-N image translation model named "Multi-CartoonGAN". She is currently a Project Assistant Professor with the Department of Information and Communication Sciences, Faculty of Science and Technology, Sophia University. Her research interests include artificial intelligence, particularly in deep learning applications, she is working on implementing various deep learning model architectures focused on tasks, such as image recognition and image generation. As part of her continuing research, she is also tackling challenges in 3D image processing, specifically focusing on deep learning models for pose estimation, and 3D modeling.

TAD GONSALVES (Member, IEEE) received the B.S. degree in theoretical physics, the M.S. degree in astrophysics, and the Ph.D. degree in information systems from Sophia University, Tokyo, Japan. He is currently a Full Professor with the Department of Information and Communication Sciences, Faculty of Science and Technology, Sophia University. He has authored or coauthored more than a hundred and fifty papers in international conferences and journals. He is the author of the book *Artificial Introduction: A Non-Technical Introduction* (2017) Sophia University Press, Tokyo, Japan, and co-author of *Artificial Intelligence for Business Optimization: Research and Applications* (2021), CRC Press, London. His research interests include bio-inspired optimization techniques and the application of deep learning techniques to diverse problems like autonomous driving, drones, digital art and music, and computational linguistics. He has also been involved in affective computing models. His research laboratory (https://www.gonken.tokyo/) in Tokyo specializes in applications of deep learning and multi-GPU computing.