Two Proposed Solutions for Mitigating Blurred Output of Autoencoder

Seongmin Kim, Kanghyun Jo
School of Electrical Engineering
Dept. of Electrical, Electronic and Computer Engineering
University of Ulsan, Ulsan, Korea
asdfhdsa1234@mail.ulsan.ac.kr, acejo@ulsan.ac.kr

***Abstract*—An autoencoder is a neural network that generates data highly similar to the input data for output. Although an autoencoder theoretically produces output almost identical to the input upon completion of learning, it actually generates blurred outputs for complex face images due to the omission of detailed information during the compression process and the use of MSE loss during learning. This paper addresses these issues by mapping detailed information from the frequency domain onto the latent space, adding to the existing latent vector, and learning using a mixed loss of MS-SSIM (Multi Scale Structural Similarity Index Measure) loss and l_1 loss instead of MSE loss. As a result, the $100 \times l_1$, $100 \times l_2$ loss, SSIM, MS-SSIM between input and output are 12, 3.1, 0.53, and 0.575 respectively, leading to the production of images of higher quality than the standard autoencoder.**

***Index Terms*—Autoencoder, Fourier Transform, SSIM**

I. Introduction

An autoencoder [1] is a unique type of neural network with imitating the input data. The encoder of the autoencoder compresses high-dimensional input data into a lower-dimensional vector, preserving only the essential information. The decoder then re-maps the compressed, lower-dimensional vector back into its original high-dimensional space. The objective of the encoder is to condense information such that the decoder can always restore it to its original form, while the aim of decoder is to reconstruct data that is as close to the original as possible, using the low-dimensional information. If all nodes in an autoencoder use linear activation functions, the machine could generate similiar result with Principal Component Analysis (PCA) [2]. Beyond this, non-linear manifold learning is feasible through non-linear activation and multilayer perceptron (MLP), and using convolution layers can also enable the construction of Convolutional Autoencoders for image. If the training was completed ideally, an autoencoder would output results identical to the input. However, as depicted in Fig. 2, considerably blurred images can be generated compared to the original input image. There are several reasons for such a phenomenon. This paper primarily examines two of those reasons. Firstly, the compression process leads to information loss. Similar to PCA, an autoencoder compresses complex high-dimensional data such as images into a lower-dimensional vector, retaining primarily the essential elements of the data. During this process, intricate details of images (such as boundaries, pupils) become significantly lost compared to the more general information about the input data (like the shape of the face, the overall pixel arrangement). Therefore, as shown in Fig. 2, the decoder effectively reconstructs the general information about the input data but fails to retain the detailed description. Secondly, it is the use of Mean Square Error Loss (MSE Loss). MSE Loss, a loss function used for predictions on continuous classes such as height and weight, is based on the negative log-likelihood of the Gaussian distribution. Due to the MSE Loss assigning equal weight to all pixels, machines are unable to make predictions that highlight specific information such as edges. This paper proposes methods for resolving these two problems.

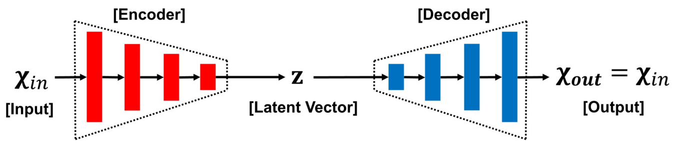

Figure 1: An illustration depicts the structure of standard autoencoder. $\mathcal{X}_{in}$ denotes the input data, $\mathcal{X}_{out}$ signifies the output data, and $\mathbf{z}$ represents the latent vector into which $\mathcal{X}_{in}$ is compressed by the encoder.

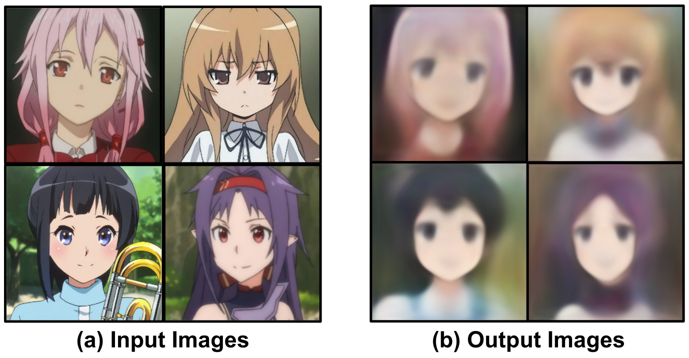

Figure 2: Illustration of the output when anime face images are processed by a Deep Convolutional Autoencoder. The output images largely resemble the input, but a distortion in the detailed areas is noticeable.

The main contributions of this paper are two-fold:

- This paper proposes a method that uses information from

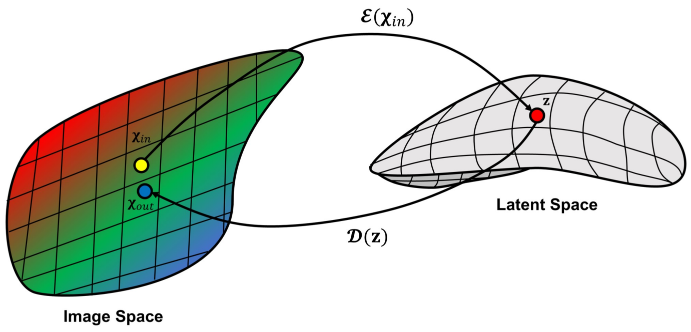

Figure 3: This figure depicts the process where an autoencoder fails to map the data properly while compressing high-dimensional images into lower dimensions and reverting them back to high dimensions. $\mathcal{X}_{in}$ and $\mathcal{X}_{out}$ represent the input and output image, $\mathbf{z}$ refers to the compressed latent vector, $\mathcal{E}$ corresponds to the encoder, and $\mathcal{D}$ represents the decoder.

the frequency domain to add detailed information in the Latent space.

- This paper proposes a method to enhance the quality of output images from autoencoders by utilizing SSIM-based mix loss [3].

II. Background

A. Frequency Domain Analysis on Image Signal

The fourier transform is a mathematical method. It can transform a time/spatial domain signal to frequency domain. The frequency domain signal is a complex signal. So, it consists magnitude and phase spectrum. The comprehensive shape of object is consisted in magnitude spectrum. The detail pattern of image is consisted in phase spectrum. When the image is transformed to the frequency domain, the origin is located in the left top corner. This phenomenon can make analyzing the image difficult. Therefore, the origin point of transformed image is generally shifted to center for the image processing such as Fig. 4. The Fig. 5 is fourier transform

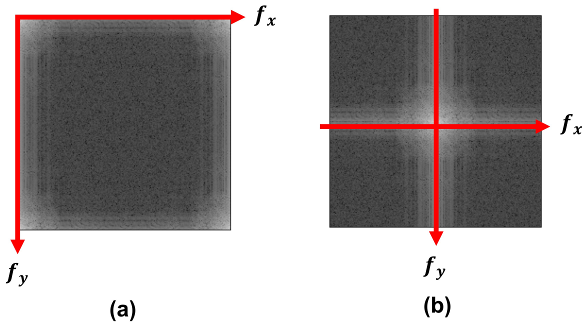

Figure 4: The origin point is shifted left-top to center of image. (a) is before origin shifting and (b) is after shifting.

results of face image. The following Fig. 5 (b), (c) show the magnitude and phase of the image (a). In image, the low-frequency component of complex signal usually consists of areas with uniform pixel intensity, such as background or large flat regions. The high-frequency component contains rapidly changing pixel values, such as edges or corners. According to

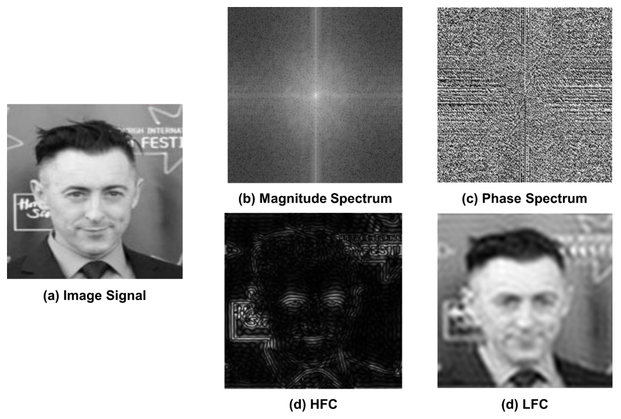

Figure 5: The fourier transform results of a face image from CELEB A dataset [4]. (a) is 2-dimensional spatial domain signal. (b) and (c) are magnitude and phase spectrum of complex signal. (d) and (e) are high frequency component and low frequency component. Where (d) and (e) are obtained by 2-dimensional high pass filter and loss pass filter.

Fig. 5, the overall shape information of the image resides in the low-frequency components, while detailed information is contained in the high-frequency components.

B. SSIM: Structural Similarity Index Measure

SSIM (Structural Similarity Index Measure) compares the similarity between two images using three factors human utilizes when recognizing images: luminance, contrast, and structure. SSIM values range between 0 and 1, with a value closer to 1 indicating a greater similarity between the two images. Luminance represents the brightness of an image and can be calculated as the average pixel value, as shown in Eq. (1).

$$\mu_x = \frac{1}{N}\sum_{i=1}^{N} x_i \tag{1}$$

x_i denotes pixel value and N represents the number of pixels. Contrast indicates the degree of change in the brightness of an image. It can be calculated as the standard deviation of pixel values, as depicted in Eq. (2).

$$\sigma_x = \sqrt{\frac{1}{N-1}\sum_{i=1}^{N}(x_i - \mu_x)^2} \tag{2}$$

Structure illustrates the structural differences in pixel values. Pixel values are re-defined by normalized pixel distribution with Eq. (3). Where $\mathcal{X}$ represents an image.

$$\frac{\mathcal{X} - \mu_x}{\sigma_x} \tag{3}$$

Differences in Luminance, Contrast, and Structure between two images can be calculated using Eq. (4) through (6).

$$l(x,y)=\frac{2\mu_x\mu_y+C_1}{\mu_x^2+\mu_y^2+C_1} \tag{4}$$

$$c(x,y)=\frac{2\sigma_x\sigma_y+C_2}{\sigma_x^2+\sigma_y^2+C_2} \tag{5}$$

$$s(x,y)=\frac{\sigma_{xy}+C_3}{\sigma_x\sigma_y+C_3} \tag{6}$$

σ_{xy} represents the covariance between two images x and y, and C_1, C_2, C_3 are constants, valued at 6.5025, 58.5225, and $\frac{C_2}{2}$, respectively. Values closer to 1 in Eq. (4) to (6) signify that the luminance, contrast, and structure components of the two images are similar. SSIM, requiring the simultaneous consideration of luminance, contrast, and structure, can be represented as shown in Eq. (7). With alpha, beta, and gamma all set to 1, the equation can be represented to appear as shown in Eq. (8) to (10).

$$\mathrm{SSIM}(x,y)=l(x,y)^{\alpha}c(x,y)^{\beta}s(x,y)^{\gamma} \tag{7}$$

$$\mathrm{SSIM}(x,y)=l(x,y)c(x,y)s(x,y) \tag{8}$$

$$=\frac{2\mu_x\mu_y+C_1}{\mu_x^2+\mu_y^2+C_1}\frac{2\sigma_x\sigma_y+C_2}{\sigma_x^2+\sigma_y^2+C_2}\frac{\sigma_{xy}+\frac{C_2}{2}}{\sigma_x\sigma_y+\frac{C_2}{2}} \tag{9}$$

$$=\frac{(2\mu_x\mu_y+C_1)(2\sigma_{xy}+C_2)}{(\mu_x^2+\mu_y^2+C_1)(\sigma_x^2+\sigma_y^2+C_2)} \tag{10}$$

III. Proposed Method

A. Model Architecture

This study proposes a modified architecture of deep convolutional autoencoder, as illustrated in Fig. 6, designed to address the limitations of traditional autoencoders in compressing information insufficiently. Upon receiving image $\mathcal{X}$ as input, the process bifurcates into two distinct paths: one path feeds the image directly into the encoder, while the other path transforms the image into the frequency domain through Fourier transformation. The signal in the frequency domain subsequently splits into two branches: one branch calculates the Phase Spectrum, while the other branch extracts the high-frequency components of the image via a high pass filter. The phase information and high-frequency information of the image represent details pertaining to intricate patterns and forms within the image. Adding these information elements to the existing compressed latent vector enriches the information of the latent vector. However, due to the high-dimensionality of both phase and high-frequency components, a mapping onto the lower-dimensional latent space becomes necessary. Hence, the two concatenated information segments are mapped into a single rich information set through the Information Mapping Network. The compressed information is added to the latent vector. The possibility of mathematical operations between latent vectors has been proven in this paper [5]. The synthesized latent vector is remapped into an image through the decoder.

B. SSIM based Training Approach

This paper employs the SSIM-based loss function proposed in [3] to mitigate the blurring phenomenon in output images of the autoencoder compared to the originals. The [3] proposed additional losses such as MS-SSIM (Multi Scale SSIM) loss and MS-SSIM + l_1 loss, in addition to SSIM loss. According to the [3], while the l_2 loss penalizes large errors and is tolerant to small ones, the l_1 loss, which does not over-penalize large errors, is can be considered more advantageous. Furthermore, [3] demonstrated experimentally that switching between l_2 loss and l_1 loss during training reduces the overall loss. On the basis of these preceding research outcomes, this paper trained the model via the following switched mixed loss.

$$\mathcal{L}=\begin{cases}\alpha\mathcal{L}^{\mathrm{MS-SSIM}}+(1-\alpha)\mathcal{L}^{l_1}, & \text{Epoch}\le\beta\\ \alpha\mathcal{L}^{\mathrm{MS-SSIM}}+(1-\alpha)\mathcal{L}^{l_2}, & \text{Epoch}>\beta\end{cases} \tag{11}$$

α is a hyperparameter that balances between the two losses, and β signifies the epoch at which to switch. l_1 and l_2 losses are expressed as per Eq. (12) and (13), respectively, while the MS-SSIM Loss is depicted in Eq. (14).

$$\mathcal{L}^{l_1}=\frac{1}{N}\sum_{i=1}^{N}|x_i-\hat{x}_i| \tag{12}$$

$$\mathcal{L}^{l_2}=\frac{1}{N}\sum_{i=1}^{N}(x_i-\hat{x}_i)^2 \tag{13}$$

$$\mathcal{L}^{\mathrm{MS-SSIM}}=1-l_M^{\alpha}\cdot\prod_{j=1}^{M}cs_j^{\beta_j} \tag{14}$$

Where, x_i and $\hat{x}_i$ represent the i-th pixel of the input and output images respectively, and N signifies the total number of pixels. M refers to the highest scale among multiple stages of changing scales. This study performed experiments with various loss combinations, details of which are elaborated in the experiments section.

IV. Experiment

A. Datasets

Selfie2Anime dataset is created for image-to-image translation between selfie images and anime face images. For this paper, only the anime face images were extracted and utilized from the selfie2anime dataset [6]. A total of 3,500 images were used for the experiments, with 3,000 for training, 400 for validation, and 100 for testing.

B. Experiment Setup

1) Equipment: The experiments were conducted using the following equipment.

- **CPU**: Intel(R) Xeon(R) Gold 5218R CPU @ 2.10GHz 1EA
- **GPU**: NVIDIA A100 PCIe 40GB 4EA
- **RAM**: SAMSUNG DDR4 64GB @ 2.933Ghz 8EA

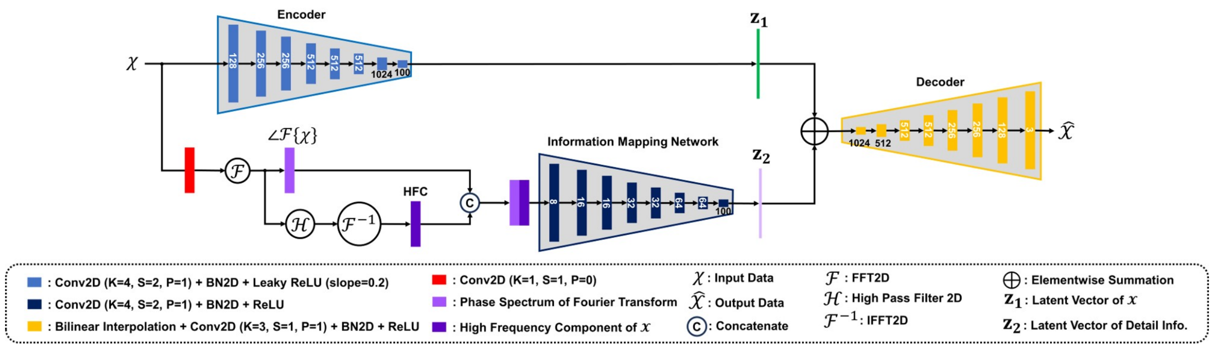
Encoder
$\mathcal{X}$
128
256
256
512
512
512
1024
100
$\mathbf{z_1}$
Decoder
1024
512
512
512
256
128
3
$\widehat{\mathcal{X}}$
$\angle\mathcal{F}\{\mathcal{X}\}$
$\mathcal{F}$
$\mathcal{H}$
$\mathcal{F}^{-1}$
HFC
C
Information Mapping Network
8
16
16
32
32
64
64
100
$\mathbf{z_2}$
: Conv2D (K=4, S=2, P=1) + BN2D + Leaky ReLU (slope=0.2)
: Conv2D (K=4, S=2, P=1) + BN2D + ReLU
: Bilinear Interpolation + Conv2D (K=3, S=1, P=1) + BN2D + ReLU
: Conv2D (K=1, S=1, P=0)
: Phase Spectrum of Fourier Transform
: High Frequency Component of $\mathcal{X}$
$\mathcal{X}$: Input Data
$\widehat{\mathcal{X}}$: Output Data
C : Concatenate
$\mathcal{F}$: FFT2D
$\mathcal{H}$: High Pass Filter 2D
$\mathcal{F}^{-1}$: IFFT2D
$\oplus$: Elementwise Summation
$\mathbf{z_1}$: Latent Vector of $\mathcal{X}$
$\mathbf{z_2}$: Latent Vector of Detail Info.

Figure 6: This figure depicts proposed architecture of model.

2) Optimizer: This study utilized the Lion Optimizer [7], which outperforms AdamW [8]. In all experiments, β_1 and β_2 were set at 0.9 and 0.999, respectively, with no weight decay employed. ϵ was set at 1e-8.

3) Learning Rate Scheduler: The ReduceLROnPlateau, built into Pytorch [9], was employed as the learning rate scheduler. Should the validation loss exceed the current lowest validation loss for a specified patience epoch continuously, the learning rate decreases by a predetermined factor. For this experiment, the patience and factor were set at 3 and 0.9, respectively.

4) Standard Autoencoder: The structure of the standard deep convolutional autoencoder used in the experiment aligns with Fig. 7 below.

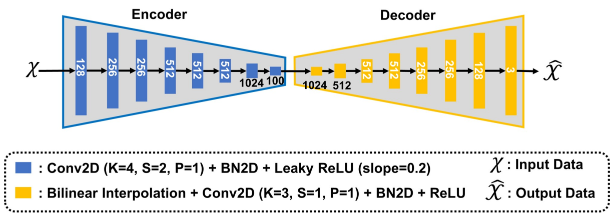
Encoder
Decoder
$\mathcal{X}$
128
256
256
512
512
512
1024
100
1024
512
512
512
256
256
128
3
$\widehat{\mathcal{X}}$
: Conv2D (K=4, S=2, P=1) + BN2D + Leaky ReLU (slope=0.2)
: Bilinear Interpolation + Conv2D (K=3, S=1, P=1) + BN2D + ReLU
$\mathcal{X}$: Input Data
$\widehat{\mathcal{X}}$: Output Data

Figure 7: An illustration depicts the architecture of standard deep convolutional autoencoder.

C. Experiment Result

1) Comparison: Both the Original and Proposed models were trained uniformly with 300 epochs, an initial learning rate of 1×10^{-4}, a batch size of 128, and input image resolution of 256×256. Here, the hyperparameters for the proposed model, such as α, were set at 0.64, β at 100, and the cut-off frequency of the high pass filter at 20. Per Table 1, the l_1 and l_2 losses for

Table 1: Comparison result between standard autoencoder and proposed model

Model	$100 \times l_1$	$100 \times l_2$	**SSIM**	**MS-SSIM**
Original	**11.3**	**2.6**	0.518	0.521
Proposed	12	3.1	**0.53**	**0.575**

the proposed model exceed those of the original; however, the SSIM and MS-SSIM indices present an increase of 2.32% and 10.36%, respectively. Fig. 8 below depicts the outputs from the standard model and proposed model when applied to the input image. An increase in clarity is noticeable by naked eye in the output from the proposed model, which also has higher SSIM and MS-SSIM indices. The larger l_1 and l_2 losses of the proposed model result from the fact that the SSIM and MS-SSIM are metrics derived from grayscale images; hence, training based on these metrics can increase shape accuracy, but not the accuracy of color information.

2) Effect of hyperparameter: This experiment investigates the impacts of hyperparameters α, β, and cut-off frequency on the model. Tables 2, 3, and 4 respectively document the metrics resulting from changes to α, β, and cutoff frequency. Where, the cutoff frequency signifies a distance of 20 pixels from the origin. Upon first examining the influence of alpha, it becomes evident that if α becomes too small, increasing the weight of the l_1, l_2 loss, the loss for these factors reduces, but SSIM, MS-SSIM decrease. Conversely, if alpha becomes too large, l_1, l_2 losses escalate. Hence, for maintaining balance between l_1, l_2 and MS-SSIM loss, an α value around 0.5 appears suitable. Upon examination of the influence of β, as proven in [3], models trained using l_1 resulted in smaller l_2 loss than those trained using l_2. Therefore, it becomes evident that β does not significantly affect performance enhancement. Lastly, looking at the cut-off frequency (f_c), it becomes apparent that performance diminishes if it is either smaller or larger than 20 pixels distance.

Table 2: Evaluation result about proposed model with different $alpha$

α	$100 \times l_1$	$100 \times l_2$	**SSIM**	**MS-SSIM**
0.35	**11.5**	**2.8**	0.529	0.56
0.5	11.6	2.9	**0.531**	0.574
0.64	12	3.1	0.53	0.575
0.82	12.8	3.3	0.527	**0.579**

V. Conclusion

This paper introduced methods to overcome the blurred output of conventional autoencoders by compressing frequency domain information into the latent space and adding it to the existing latent vector, where detailed information is lost, and by enhancing the quality of the output image through a mixed

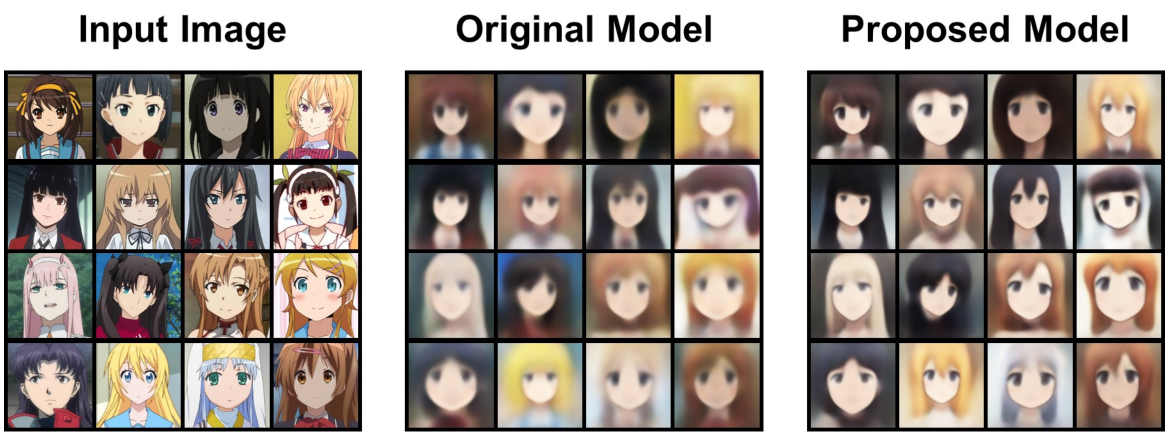

Figure 8: An illustration of models outputs.

Table 3: Evaluation result about proposed model with different $beta$

β	$100 \times l_1$	$100 \times l_2$	**SSIM**	**MS-SSIM**
inf (No Switch)	**11.9**	**3**	**0.53**	0.574
150	12	3.1	0.529	0.574
100	12	3.1	**0.53**	**0.575**

Table 4: Evaluation result about proposed model with different f_c. Where f_c denotes cut-off frequency of high pass filter.

	f_C	$100 \times l_1$	$100 \times l_2$	**SSIM**	**MS-SSIM**
$\alpha = 0.64$	10	14.8	4	0.509	0.538
	20	**12**	**3.1**	**0.53**	**0.575**
	30	**12**	3.2	0.529	**0.575**
	f_C	$100 \times l_1$	$100 \times l_2$	**SSIM**	**MS-SSIM**
$\alpha = 0.82$	10	**12.8**	3.4	0.526	0.577
	20	**12.8**	**3.3**	**0.527**	**0.579**
	30	12.9	3.4	0.522	0.549

loss of MS-SSIM loss and l_1 loss. Through experiments, the proposed methodology proved to generate images of superior quality compared to traditional autoencoders, and explored the impact of α, β, and f_c on the model. Future research will investigate loss functions utilizing metrics other than SSIM, which is incapable of measuring color information.

Acknowledgment

This result was supported by "Regional Innovation Strategy (RIS)" through the National Research Foundation of Korea(NRF) funded by the Ministry of Education(MOE)(2021RIS-003)

References

[1] Geoffrey E Hinton and Ruslan R Salakhutdinov. Reducing the dimensionality of data with neural networks. *science*, 313(5786):504–507, 2006.

[2] Dor Bank, Noam Koenigstein, and Raja Giryes. Autoencoders. *arXiv preprint arXiv:2003.05991*, 2020.

[3] Hang Zhao, Orazio Gallo, Iuri Frosio, and Jan Kautz. Loss functions for image restoration with neural networks. *IEEE Transactions on Computational Imaging*, 3(1):47–57, 2017.

[4] Ziwei Liu, Ping Luo, Xiaogang Wang, and Xiaoou Tang. Deep learning face attributes in the wild. In *Proceedings of International Conference on Computer Vision (ICCV)*, December 2015.

[5] Alec Radford, Luke Metz, and Soumith Chintala. Unsupervised representation learning with deep convolutional generative adversarial networks. *arXiv preprint arXiv:1511.06434*, 2015.

[6] Junho Kim, Minjae Kim, Hyeonwoo Kang, and Kwanghee Lee. U-gat-it: Unsupervised generative attentional networks with adaptive layer-instance normalization for image-to-image translation. *arXiv preprint arXiv:1907.10830*, 2019.

[7] Xiangning Chen, Chen Liang, Da Huang, Esteban Real, Kaiyuan Wang, Yao Liu, Hieu Pham, Xuanyi Dong, Thang Luong, Cho-Jui Hsieh, et al. Symbolic discovery of optimization algorithms. *arXiv preprint arXiv:2302.06675*, 2023.

[8] Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. *arXiv preprint arXiv:1711.05101*, 2017.

[9] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, et al. Pytorch: An imperative style, high-performance deep learning library. *Advances in neural information processing systems*, 32, 2019.