LIIF-GAN: Learning Representations With Local Implicit Image Function and GAN for Realistic Images on a Continuous Scale

Jun Seok Kang[1,2]
js.kang@ust.ac.kr
Sang Chul Ahn[1,2]
asc@kist.re.kr

[1] KIST school
University of Science and Technology
Seoul, Republic of Korea

[2] Artificial Intelligence and Robotics Institute
Korea Institute of Science and Technology
Seoul, Republic of Korea

Abstract

Recently, the Local Implicit Image Function (LIIF) has been proposed, which can generate continuous 2D image representation for pixel-based images. The continuous image representation can be presented at any resolution. However, the LIIF representation has limited fidelity when presented at higher resolution, resulting in unrealistic images. To solve this problem, simply adding a GAN can produce a realistic image, but it degrades the local structure of the image. In this paper, we propose the LIIF-GAN, a novel architecture-based deep model, to generate realistic images at continuous scales while maintaining local image structures. It utilizes a generative adversarial network (GAN) and multiple decoders for encoder features at different levels. We show that the LIIF-GAN can generate a more realistic continuous image representation than previous methods. Furthermore, we show that our new architecture retains the local image structure better than simply using a GAN with the existing architecture. The performance of the proposed method is demonstrated qualitatively and quantitatively through various experiments.

1 Introduction

The real visual world is continuous. However, when we are processing a visual scene, discrete 2D images are used due to the limitations of the computer's storage and display formats. So, if we want to use various sized images, we need to use image resizing. However, image resizing usually degrades image quality and may lead to the wrong result in some applications. To overcome this limitation, a local implicit image function (LIIF)[4] was proposed to learn continuous image representation. The LIIF is based on a neural implicit function[5, 10, 19, 22, 24, 25] that is used in many 3D vision applications.

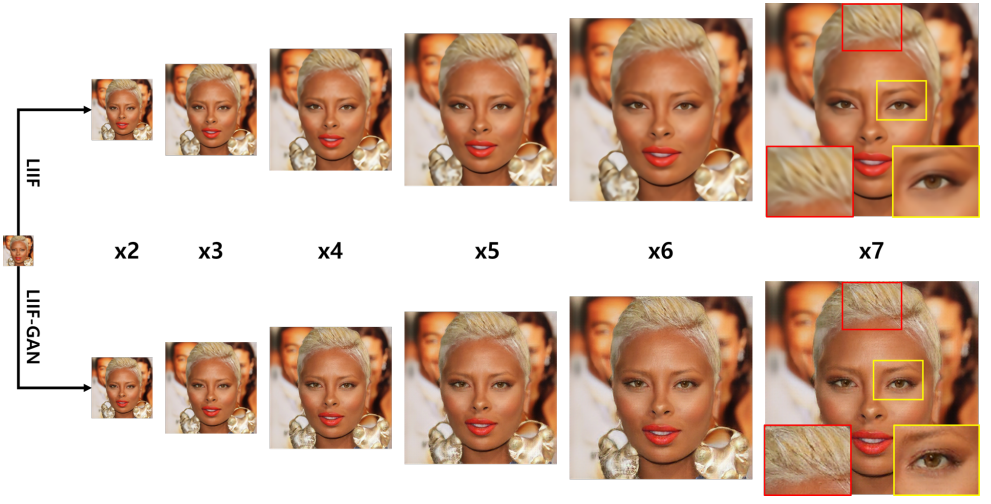

Figure 1: The Local Implicit Image Function-GAN (LIIF-GAN) represents a realistic image on a continuous scale. As shown in the figure, the LIIF-GAN can make more realistic images than the state-of-the-art LIIF can.

However, the LIIF uses only the L1 loss in the training process. According to previous studies[2, 29], training the image reconstruction model using only L1 loss (or L2 loss) degrades the fidelity of the output image. As a result, we get an image in which the local details have disappeared. So, the LIIF can't adapt to various applications that demand realistic images. The introduction of GAN[9] and perceptual loss[11] to the LIIF is an option for the above problem. However, while this method can increase the image fidelity, it can degrade the structure of the image component[2, 29]. Therefore, it is not a suitable solution to simply add GAN and perceptual loss to the model.

To solve this problem, we propose the LIIF-GAN, a new method to learn realistic image representations on a continuous scale while maintaining the structure of image components. To make the LIIF-GAN represent well-structured realistic images, multiple decoders are introduced for the reconstruction part. The first decoder is trained using L1 loss only, like the original LIIF. As a result, the first decoder outputs an unrealistic image that has well preserved internal structure. The second decoder is trained to improve the fidelity of the image. The combination of the first and second decoder's outputs is fed to GAN and perceptual loss computation. Also, we use different layer features of an encoder for the input of each decoder. As a result, our novel structure can generate well-structured, realistic images. A detailed explanation of the model is described in Sec 3.3. Fig.1 shows our results compared to the LIIF, which is the previous state-of-the-art(SOTA) method. Our contribution can be summarized as follows: 1) We introduce GAN and perceptual loss to the LIIF to increase the fidelity of the image. 2) A new model architecture utilizing two decoders is proposed to maintain the internal image structure and increase the fidelity of the image at the same time. 3) To further improve the performance, we suggest using different layer features of the encoder for the input of each decoder.

2 Related works

Super resolution Our goal is representing a realistic image on a continuous scale. If we can make the continuous image representation, then we can generate an arbitrary scale high resolution image from a low resolution image. So, we can measure the performance of representing the realistic image on a continuous scale through a super resolution task[3, 7, 14, 16, 17, 27, 28, 29, 32, 34]. Dong et al.[7] achieved high performance in super resolution by using deep convolutional neural networks. Ledig et al.[16] demonstrated that using GAN can allow us to achieve realistic results. Wang et al.[29] showed that mixing up the model parameters that are trained by L1 loss and GAN loss gets better results than using parameters that are trained with a single loss. Wang et al.[30] improved the super resolution performance by using augmented inputs, which are made by various kernels. Although there are many studies about super resolution, most of the studies use the convolutional neural network(CNN)[15] to solve the super resolution problem. Because CNN usually has a fixed input and output size, they have the limitation that they can perform super resolution with a fixed scale only. For learning realistic image representations on a continuous scale, we use the implicit function instead of the CNN.

Neural Implicit Function in 2D image representation Research using implicit functions to represent 2D images has been conducted[4, 20, 26, 31]. Stanley et al.[26] represented 2D images using a compositional pattern producing network. Chen and Zhang[5] tried to learn a latent space for simple 2D digits. Sitzmann et al.[25] found that using a periodic activation function instead of ReLU[21] can improve the ability to represent the fine details of a natural image. Chen et al.(LIIF)[4] tried to learn the space that can represent various natural images in continuous resolution. Like [6, 8, 10, 23, 24], the LIIF utilizes local latent codes. Local latent codes are used to recover complex images in the LIIF. The UltraSR[31] is follow-up research to the LIIF. It demonstrated that using spatial encoding on input coordinates can improve the performance of the LIIF. However, in both studies, the L1 loss is only used in training. Training using only L1 loss degrades image fidelity[2, 29]. Compared with them, we can get a more realistic image representation by introducing a novel network architecture to accommodate GAN and perceptual loss.

3 Method

3.1 Local Implicit Image Function

Since the LIIF is the baseline of our research, we briefly explain the LIIF. The LIIF takes a coordinate and its nearby features as inputs and generates a single RGB value. The coordinate can be a real value. The LIIF can create images of any size by repeating this process along all pixel coordinates. Let I denote an image and M represent a 2D feature map that corresponds to I. In the LIIF, an encoder computes M from I. z denotes a feature (we call it the latent code in other parts of the paper) in M. In the LIIF, the feature map M is positioned in [0, 1]x[0, 1] 2D space. Then the coordinates of each feature z can be determined according to its location in M. f_θ (with θ as its parameters) is a multilayer perceptron (MLP) based decoder shared by all images. Let's denote x as the 2D coordinate of the target image pixel. The output RGB value for the corresponding coordinate is denoted by s. The simplest version of the LIIF decoder is formulated as

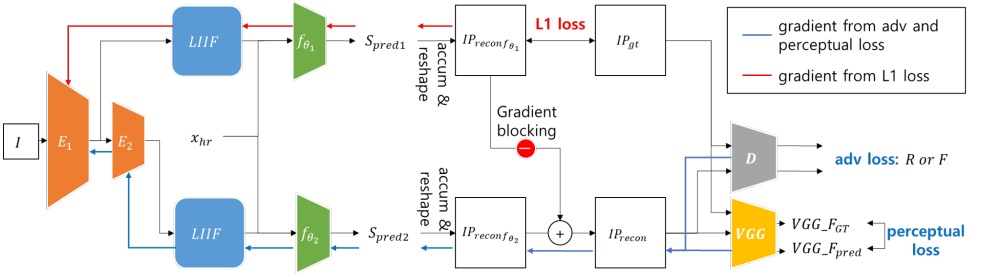

Figure 2: The architecture of the LIIF-GAN. The detail explanations of each part are described in Sec. 3.

$$s = \sum_{t \in \{00,01,10,11\}} \frac{S_{t'}}{S} f_\theta(z_t^*, [x - v_t^*, c]), \tag{1}$$

where z^* is the nearest z from x and v^* is the coordinate of z^*. Using a feature map and a coordinate as inputs, the function f_θ generates a RGB value. The 00, 01, 10, and 11 indicate top-left, top-right, bottom-left, and bottom-right directions, respectively. $S_{t'}$ is the area of the rectangle between x and $v_{t'}^*$, where t' is diagonal to t (for example, if t is 01, t' is 10). S is the sum of S_t. The c is a vector that consists of the height and width of the query pixel. $[x - v^*, c]$ means a concatenation of $x - v^*$ and c. Similar to the neural implicit function, the LIIF takes the coordinates and local latent code and then generates a continuous image representation. In Fig.3, we visualize the components of the LIIF.

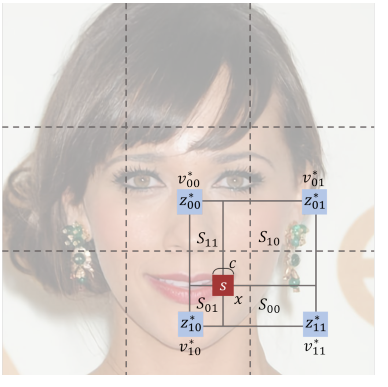

Figure 3: The components of LIIF(also used in LIIF-GAN). The detail explanations are given in Sec. 3.

The training of the LIIF is conducted as follows: 1) Prepare a ground truth image. 2) Make a down-scaled image using bicubic interpolation. The scales are between x1.0 and x4.0. 3) Create a 2D feature map of the down-scaled input image using the encoder. 4) Select some coordinates randomly from the ground truth image. 5) Compute s at each coordinate using the decoder f_θ, 2D feature map M, the target coordinate x, area S, and cell size c. 6) Calculate the L1 distance between the RGB output s and the ground truth color value. 7) Use an optimizer to reduce the distance.

3.2 Toward the Realistic Image

According to previous research [2, 29], using only L1 loss in training can lead to unrealistic image reconstruction. To improve the fidelity, GAN and perceptual loss are normally used. Since the LIIF adopts random coordinate sampling and generates a single individual RGB value, GAN and perceptual loss cannot be applied directly. It is necessary to change the sampling method. Instead of random coordinate sampling, we use random group sampling(RGS). RGS consists of two steps: 1) A random sample point in a target image is

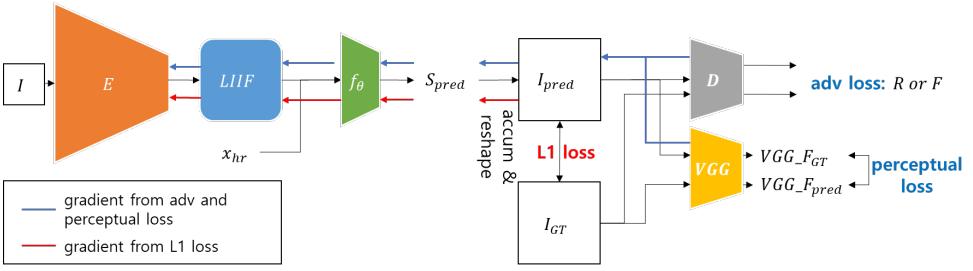

Figure 4: The architecture of the LIIF-GAN-S. The detail explanations are described in Sec. 3.2.

chosen at random. 2) Around that point, more coordinates are selected in a patch shape. Note that the initial sample coordinate and surrounding coordinates can be real-valued ones. RGB values are computed at those sample coordinates. By aggregating the RGB values, we can make an image patch. After adopting random group sampling, we can attach GAN and perceptual loss to the LIIF. We denote this model as the LIIF-GAN-S (simple). We denote the structure of the LIIF-GAN-S in Fig.4. The loss function for the LIIF-GAN-S is

$$\begin{aligned} L_{LIIF-GAN-S} = |IP_{gt} - IP_{recon}| + \log(D(IP_{gt})) \\ + \log(1 - D(IP_{recon})) + |VGG(IP_{gt}) - VGG(IP_{recon})|, \end{aligned} \tag{2}$$

where IP_{gt} is the ground truth image patch and D is the discriminator. VGG is the VGG19 feature extractor and

$$IP_{recon} = Reshape(f_\theta(z_k^*, [x_k - v_k^*, c_k])_{k \in iRGS}), \tag{3}$$

where *Reshape* is a reshape function that converts a set of RGB values into an image patch. k is the sampling index, and *iRGS* is the set of random ground sampling indexes.

3.3 Maintaining the Internal Structure of Image Components

The introduction of GAN and perceptual loss can enable the production of a realistic image representation. However, unfortunately, it corrupts the internal structure of image components. It is known that L1 loss can learn the internal structure of image components better than GAN or perceptual loss[2, 29]. To address the problem, we suggest LIIF-GAN. The structure of the LIIF-GAN is shown in Fig.2. In the LIIF-GAN, we divide the reconstruction part into two by using two decoders. In the first part, a decoder is trained by L1 loss only. The loss function for the first decoder is

$$L_{D_1} = |IP_{gt} - IP_{recon\ f_{\theta_1}}|, \tag{4}$$

where $IP_{recon\ f_{\theta_1}}$ means the image patch from the first decoder. As a result, the output of the first decoder can preserve the internal structure of image components well. In the second part, a decoder generates incremental RGB values for fine details. The outputs of the first and second decoders are added together and fed to the GAN and perceptual loss computation module. Gradient blocking is used so that the gradient calculated by the GAN and the perceptual loss cannot affect the first decoder. The loss function for the second decoder is

$$\begin{aligned} L_{D_2} = \log(D(IP_{gt})) + \log(1 - D(IP_{recon\ f_{\theta_1}} + IP_{recon\ f_{\theta_2}})) \\ + |VGG(IP_{gt}) - VGG(IP_{recon\ f_{\theta_1}} + IP_{recon\ f_{\theta_2}})|. \end{aligned} \tag{5}$$

As a result, the second decoder can only learn how to improve the fidelity of the images. Using the two decoders, the internal image structure training and image fidelity training do not disturb each other.

Additionally, we use the features from different layers of the encoder for each decoder to enrich the diversity of input features. We use mid-layer features for the first decoder and last-layer features for the second decoder, respectively. The final loss function for the LIIF-GAN is

$$L_{LIIF-GAN} = L_{D_1} + L_{D_2}, \tag{6}$$

which is minimized for training the model.

4 Experiments

4.1 Evaluation

A continuous representation can be expected to have infinite precision and can represent an arbitrary high-resolution image. Thus, we evaluate the performance of the model through the super resolution task. For quantitative evaluation, the Peak Signal-to-Noise Ratio (PSNR), Structural SIMilarity (SSIM), and Learned Perceptual Image Patch Similarity (LPIPS)[33] are used. The PSNR and SSIM are suitable for measuring structural similarity. In contrast, the LPIPS is more suitable for measuring the fidelity of an image.

Two datasets are used for experiments. One is the DIV2K dataset[1], and the other is the CelebA-HQ dataset[12]. The DIV2K consists of 1000 images in 2K resolution. We follow the original division of the DIV2K dataset. Training is done on the DIV2K training set with 800 images. For testing, we use the DIV2K validation set with 100 images, which is the same as in the previous paper[4]. The CelebA-HQ dataset consists of over 30,000 people's faces selected from the CelebA dataset[18]. The resolution of each image is 1024x1024. We use 25,000 images for training and 5,000 images for testing.

4.2 Implementation details

For the encoder, we used the EDSR[17] structure as a baseline, like in the previous paper[4]. The sizes of inputs are 48x48 and 64x64 for the DIV2K and CelebA-HQ experiments, respectively. The two decoders have the same structure that consists of a 5-layer MLP with ReLU activation and 256 hidden nodes in each layer. The output is a single RGB value. For the discriminator, we used a CNN model. We set the size of the patch that can be made by random group sampling equal to the input size. A pretrained VGG19 network is used to calculate the perceptual loss. We used the Adam[13] optimizer with an initial learning rate 10^{-4}. The learning rate decays by a factor of 0.5 every 200 epochs. The experimental setting of the LIIF is the same as the LIIF-GAN, except for the model architecture. For previous fully convolutional neural networks(FCNN) based methods (EDSR, ESRGAN, and Real-ESRGAN), we followed previous papers' implementations (refer to [17], [29], and [30], respectively).

4.3 Quantitative and Qualitative Comparisons with Prior Works

Training and Test setting During the training on the DIV2K dataset, the target is the cropped original image and the input is the down-scaled image with various scale fac-

Table 1: The PSNR, SSIM, and LPIPS scores on the DIV2K dataset. The up and down arrows mean that the higher and lower scores are better, respectively. N/A means not available. The bold texts denote the best scores. The texts in red indicate the best scores among the models adopting GAN. Note that there is a perception-distortion tradeoff[2], meaning that adoption of GAN can degrade the PSNR/SSIM scores.

Upscale		x2	x3	x4	x4.5	x5	x5.5	x6	x6.5	x7
Bicubic	PSNR↑	31.0366	28.2531	26.6941	26.1249	25.6483	25.2262	24.8671	24.5336	24.2403
	SSIM↑	0.9012	0.8274	0.7657	0.7398	0.7170	0.6964	0.6790	0.6628	0.6489
	LPIPS↓	0.1436	0.2624	0.3407	0.3708	0.3989	0.4244	0.4463	0.4686	0.4878
EDSR(x4)	PSNR↑	N/A	N/A	**29.2875**	26.2120	25.5856	25.2278	24.9007	24.5659	24.2661
	SSIM↑	N/A	N/A	**0.8402**	0.7509	0.7234	0.7016	0.6828	0.6655	0.6507
	LPIPS↓	N/A	N/A	0.1933	0.2988	0.3308	0.3806	0.4212	0.4521	0.4760
ESRGAN(x4)	PSNR↑	N/A	N/A	25.4396	23.9158	23.9467	23.1217	23.4173	22.5057	21.8826
	SSIM↑	N/A	N/A	0.7336	0.6920	0.6913	0.6640	0.6683	0.6356	0.6124
	LPIPS↓	N/A	N/A	0.1062	0.2658	0.3364	0.3921	0.4310	0.4670	0.4956
Real-ESRGAN(x4)	PSNR↑	N/A	N/A	24.7586	24.4056	24.1763	23.9371	23.6608	23.3091	22.9488
	SSIM↑	N/A	N/A	0.7083	0.6885	0.6776	0.6651	0.6511	0.6348	0.6203
	LPIPS↓	N/A	N/A	0.2055	0.2231	0.2297	0.2409	0.2540	0.2646	0.2732
LIIF	PSNR↑	**34.6661**	**30.9782**	29.0269	**28.3275**	**27.7319**	**27.2199**	**26.7880**	**26.3909**	**26.0445**
	SSIM↑	**0.9431**	**0.8864**	0.8339	**0.8104**	**0.7887**	**0.7685**	**0.7505**	**0.7340**	**0.7191**
	LPIPS↓	0.0596	0.1416	0.2025	0.2251	0.2463	0.2673	0.2865	0.3033	0.3183
LIIF-GAN-S	PSNR↑	32.2442	27.9404	26.2872	25.6141	25.0529	24.6961	24.3560	24.0136	23.7083
	SSIM↑	0.9215	0.8173	0.7528	0.7228	0.6949	0.6739	0.6541	0.6353	0.6176
	LPIPS↓	0.0251	0.0634	0.1017	0.1180	0.1350	0.1510	0.1673	0.1829	0.1993
LIIF-GAN-SF	PSNR↑	32.1692	28.2658	26.3703	25.6986	25.1800	24.7327	24.3701	24.0206	23.7445
	SSIM↑	0.9102	0.8248	0.7533	0.7227	0.6969	0.6725	0.6520	0.6329	0.6163
	LPIPS↓	0.0252	**0.0614**	0.1026	0.1186	**0.1347**	0.1535	0.1701	0.1870	0.2002
LIIF-GAN	PSNR↑	32.2508	28.6562	26.4465	25.8676	25.4848	24.9758	24.6364	24.3327	24.1331
	SSIM↑	0.9131	0.8344	0.7555	0.7295	0.7091	0.6832	0.6638	0.6466	0.6339
	LPIPS↓	**0.0242**	0.0651	**0.0996**	**0.1149**	0.1351	**0.1476**	**0.1641**	**0.1807**	**0.1959**

tors(from x1.0 to x0.25). The model is trained for 1000 epochs with a batch size of 128 and 20 repetitions. In the test, the target is the original image and the input is the resized one with various scale factors. The scale factors are x2, x3, x4, x4.5, x5, x5.5, x6, x6.5, and x7. During the training on the CelebA-HQ dataset, the target is the 256x256 image and the input is the down-scaled one with various scale factors(from x1.0 to x0.25). The model is trained for 200 epochs with a batch size of 128 and 20 repetitions. In the test, the input is a 64x64 image and the target is a scaled image. The scale factors are x2, x3, x4, x4.5, x5, x5.5, x6, x6.5, and x7. Note that the FCNN based methods support a fixed scale super resolution only, so the input image is resized with a bicubic algorithm before the super resolution process. For example, if we want to recover a 320x320(=64x5) target from a 64x64 input using the FCNN based model, the input is resized to 80x80(=320/4) via a bicubic algorithm.

Quantitative Comparisons Table 1 shows the scores of each model on the DIV2K dataset. First, the results of EDSR, ESRGAN, and Real-ESRGAN, which target fixed-scale super-resolution, show good results on the trained scale but very large performance degradation on the untrained scales. On the other hand, it can be seen that LIIF-GAN works better even on the untrained scales. This confirms that LIIF-GAN performs well on continuous scale image representation. Second, the LPIPS scores of the LIIF-GAN are superior to those of the LIIF on all scales. Since LPIPS is an indicator to measure perceptual quality, the LIIF-GAN is better for realistic image representation than the LIIF. Note that the LIIF-GAN has lower PSNR/SSIM scores than the LIIF because improving perceptual quality often leads to a sacrifice of PSNR/SSIM scores. It is well-known as a perception-distortion tradeoff[2]. Third, it

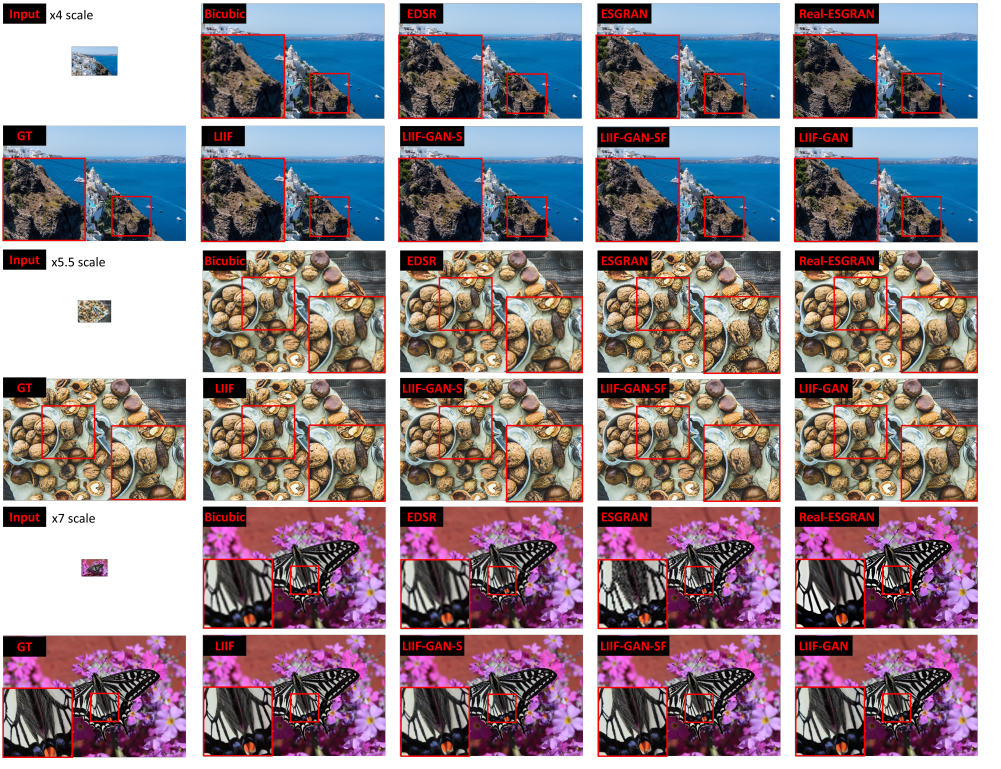

Figure 5: Qualitative comparison on the DIV2K dataset. Various scale super resolution results of each model are shown. Zoom in for a better comparison.

can be seen that the LIIF-GAN is superior to the existing GAN-related methods(ESRGAN, Real-ESRGAN). The LIIF-GAN shows better PSNR/SSIM and LPIPS scores than the ESRGAN and Real-ESRGAN on all scales. In summary, the LIIF-GAN can reconstruct realistic images better on a continuous scale than conventional methods, including the LIIF (the SOTA method). Because the results from the CelebA-HQ dataset are similar, we include a table about the results in the supplementary material due to page limitation.

Qualitative Comparisons Through Fig.5 and Fig.6, we can understand the quantitative results more clearly. The results of the EDSR and LIIF preserve the main edges well but lose the detailed textures. As a result, the output images are not realistic. On the contrary, the results of the LIIF-GAN well preserve the detailed textures. The results of ESRGAN and Real-ESRGAN are well reconstructed on the x4 scale(training scale), but reconstruction quality is significantly degraded on the out-of-training scales. Conversely, the LIIF-GAN works well on out-of-training scales.

4.4 Ablation Studies

As mentioned in Sec.3.3, in the LIIF-GAN, we use a novel structure that uses multiple decoders and encoder features. We try to show the effect of using the multiple decoders and encoder features through ablation studies. We made a model that uses only a single decoder.

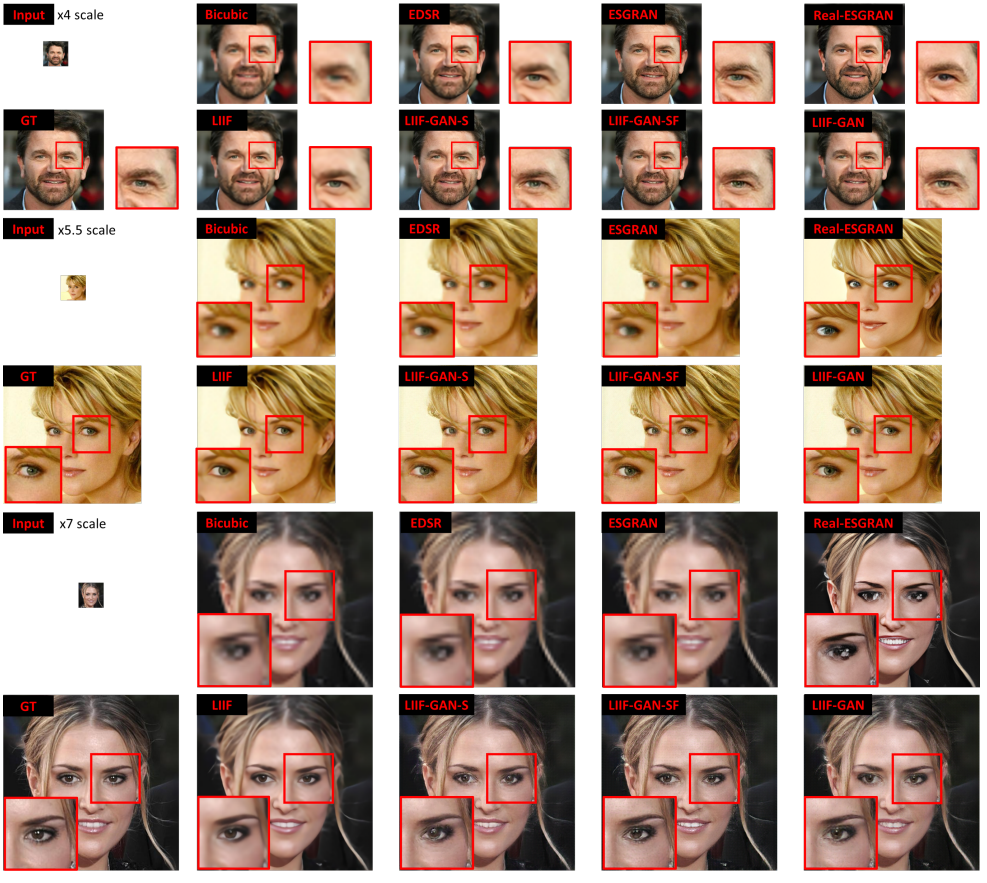

Figure 6: Qualitative comparison on the CelebA-HQ dataset. Various scale super resolution results of each model are shown. Zoom in for a better comparison.

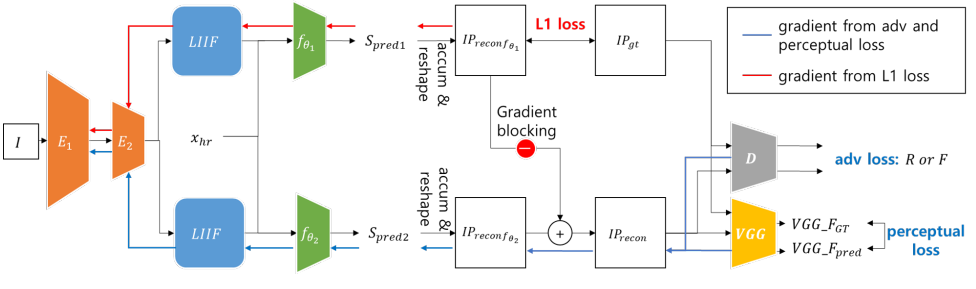

Figure 7: The architecture of the LIIF-GAN-SF. The detail explanations are described in Sec. 4.4.

We called this model the LIIF-GAN-S (Simple). We also made another model that uses the last layer feature as the common input of the multiple decoders. We called this model the LIIF-GAN-SF (Single Feature). We denote the structure of the LIIF-GAN-SF in Fig.7. We did the same experiments as the LIIF-GAN.

Table 1, Fig.5, and Fig.6 contain the results of the above ablation models. Both ablation models have worse PSNR/SSIM scores than the LIIF-GAN, while the LPIPS scores are similar. It means that the absence of the multiple decoders and encoder features degrades the local structure of images. We can see it more clearly in Fig.6. If we focus on the eyes(iris), the eyes' structure is collapsed in the results of the LIIF-GAN-S and LIIF-GAN-SF. On the contrary, the results of the LIIF-GAN preserve the eyes' structure well.

5 Limitation

The proposed LIIF-GAN has two decoders, one for reconstructing a well-structured image and the other for generating plausible fidelity. The second decoder is responsible for creating high-frequency information to be added to the first decoder's output. We found some artifacts and less plausible fidelity in the generated details of images when the scaling factor was about two times bigger than that used in the training. It is a problem that originated from the GAN because the GAN has an essential problem of instability when used outside of the learned domain. If there is another available method for fidelity training, it can be incorporated into the LIIF-GAN architecture. Finding an appropriate method for fidelity training is a candidate for future research.

6 Conclusion

This paper presents a Local Implicit Image Function–GAN (LIIF-GAN) for learning the representation of realistic images on a continuous scale. Utilizing two decoders with different layer features of an encoder, the LIIF-GAN improves the fidelity and maintains the structure of the images. We designed the architecture of the LIIF-GAN to have separate training paths for information of different frequencies. The first decoder tries to maintain the structure of images, which is related to low-frequency information. On the other hand, the second decoder generates plausible fidelity in the images, which is related to high-frequency information. With the LIIF-GAN, we can make a well-structured, realistic image representation on a continuous scale for general images. Experiments demonstrated that the LIIF-GAN could learn a more realistic image representation than the previous SOTA method (LIIF). We also showed that the new architecture of the LIIF-GAN is very effective in learning realistic image representation by comparing it with two ablated models. We think this new architecture can be adapted when applying the GAN network.

7 Acknowledgements

This work was supported by the Korea Institute of Science and Technology (KIST) Institutional Program (Project No. 2E31591).

References

[1] Eirikur Agustsson and Radu Timofte. Ntire 2017 challenge on single image super-resolution: Dataset and study. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) workshops*, pages 126–135, 2017.

[2] Yochai Blau, Roey Mechrez, Radu Timofte, Tomer Michaeli, and Lihi Zelnik-Manor. The 2018 pirm challenge on perceptual image super-resolution. In *Proceedings of the European Conference on Computer Vision (ECCV) Workshops*, pages 0–0, 2018.

[3] Hong Chang, Dit-Yan Yeung, and Yimin Xiong. Super-resolution through neighbor embedding. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, volume 1, page I. IEEE, 2004.

[4] Yinbo Chen, Sifei Liu, and Xiaolong Wang. Learning continuous image representation with local implicit image function. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 8628–8638, 2021.

[5] Zhiqin Chen and Hao Zhang. Learning implicit fields for generative shape modeling. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 5939–5948, 2019.

[6] Julian Chibane, Thiemo Alldieck, and Gerard Pons-Moll. Implicit functions in feature space for 3d shape reconstruction and completion. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 6970–6981, 2020.

[7] Chao Dong, Chen Change Loy, Kaiming He, and Xiaoou Tang. Learning a deep convolutional network for image super-resolution. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 184–199. Springer, 2014.

[8] Kyle Genova, Forrester Cole, Avneesh Sud, Aaron Sarna, and Thomas Funkhouser. Local deep implicit functions for 3d shape. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 4857–4866, 2020.

[9] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. *Advances in neural information processing systems*, 27, 2014.

[10] Chiyu Jiang, Avneesh Sud, Ameesh Makadia, Jingwei Huang, Matthias Nießner, and Thomas Funkhouser. Local implicit grid representations for 3d scenes. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 6001–6010, 2020.

[11] Justin Johnson, Alexandre Alahi, and Li Fei-Fei. Perceptual losses for real-time style transfer and super-resolution. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 694–711. Springer, 2016.

[12] Tero Karras, Timo Aila, Samuli Laine, and Jaakko Lehtinen. Progressive growing of gans for improved quality, stability, and variation. *arXiv preprint arXiv:1710.10196*, 2017.

[13] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

[14] Wei-Sheng Lai, Jia-Bin Huang, Narendra Ahuja, and Ming-Hsuan Yang. Deep laplacian pyramid networks for fast and accurate super-resolution. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 624–632, 2017.

[15] Yann LeCun, Léon Bottou, Yoshua Bengio, and Patrick Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998.

[16] Christian Ledig, Lucas Theis, Ferenc Huszár, Jose Caballero, Andrew Cunningham, Alejandro Acosta, Andrew Aitken, Alykhan Tejani, Johannes Totz, Zehan Wang, and Wenzhe Shi. Photo-realistic single image super-resolution using a generative adversarial network. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 4681–4690, 2017.

[17] Bee Lim, Sanghyun Son, Heewon Kim, Seungjun Nah, and Kyoung Mu Lee. Enhanced deep residual networks for single image super-resolution. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) workshops*, pages 136–144, 2017.

[18] Ziwei Liu, Ping Luo, Xiaogang Wang, and Xiaoou Tang. Deep learning face attributes in the wild. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 3730–3738, 2015.

[19] Lars Mescheder, Michael Oechsle, Michael Niemeyer, Sebastian Nowozin, and Andreas Geiger. Occupancy networks: Learning 3d reconstruction in function space. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 4460–4470, 2019.

[20] Alexander Mordvintsev, Nicola Pezzotti, Ludwig Schubert, and Chris Olah. Differentiable image parameterizations. *Distill*, 3(7):e12, 2018.

[21] Vinod Nair and Geoffrey E Hinton. Rectified linear units improve restricted boltzmann machines. In *International conference on machine learning*, 2010.

[22] Jeong Joon Park, Peter Florence, Julian Straub, Richard Newcombe, and Steven Lovegrove. Deepsdf: Learning continuous signed distance functions for shape representation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern (CVPR) Recognition*, pages 165–174, 2019.

[23] Songyou Peng, Michael Niemeyer, Lars Mescheder, Marc Pollefeys, and Andreas Geiger. Convolutional occupancy networks. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 523–540. Springer, 2020.

[24] Shunsuke Saito, Zeng Huang, Ryota Natsume, Shigeo Morishima, Angjoo Kanazawa, and Hao Li. Pifu: Pixel-aligned implicit function for high-resolution clothed human digitization. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 2304–2314, 2019.

[25] Vincent Sitzmann, Julien Martel, Alexander Bergman, David Lindell, and Gordon Wetzstein. Implicit neural representations with periodic activation functions. *Advances in Neural Information Processing Systems*, 33:7462–7473, 2020.

[26] Kenneth O Stanley. Compositional pattern producing networks: A novel abstraction of development. *Genetic programming and evolvable machines*, 8(2):131–162, 2007.

[27] Ying Tai, Jian Yang, Xiaoming Liu, and Chunyan Xu. Memnet: A persistent memory network for image restoration. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 4539–4547, 2017.

[28] Radu Timofte, Vincent De Smet, and Luc Van Gool. Anchored neighborhood regression for fast example-based super-resolution. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 1920–1927, 2013.

[29] Xintao Wang, Ke Yu, Shixiang Wu, Jinjin Gu, Yihao Liu, Chao Dong, Yu Qiao, and Chen Change Loy. Esrgan: Enhanced super-resolution generative adversarial networks. In *Proceedings of the European Conference on Computer Vision (ECCV) workshops*, page 0, 2018.

[30] Xintao Wang, Liangbin Xie, Chao Dong, and Ying Shan. Real-esrgan: Training real-world blind super-resolution with pure synthetic data. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV) Workshops*, pages 1905–1914, October 2021.

[31] Xingqian Xu, Zhangyang Wang, and Humphrey Shi. Ultrasr: Spatial encoding is a missing key for implicit image function-based arbitrary-scale super-resolution. *arXiv preprint arXiv:2103.12716*, 2021.

[32] Kai Zhang, Luc Van Gool, and Radu Timofte. Deep unfolding network for image super-resolution. In *Proceedings of the IEEE/CVF conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3217–3226, 2020.

[33] Richard Zhang, Phillip Isola, Alexei A Efros, Eli Shechtman, and Oliver Wang. The unreasonable effectiveness of deep features as a perceptual metric. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 586–595, 2018.

[34] Yulun Zhang, Yapeng Tian, Yu Kong, Bineng Zhong, and Yun Fu. Residual dense network for image super-resolution. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2472–2481, 2018.