Super-Resolution With Local Implicit Image Function and SIREN

Weiyun Jiang
Electrical Engineering
Stanford University
wyjiang@stanford.edu

Abstract

With the help of emerging high-performance GPUs and new neural network architectures, deep learning for super-resolution has achieved tremendous success. Local implicit image function (LIIF) can represent a 2-D image continuously. In this paper, we propose to replace the regular ReLU MLP inside LIIF with SIREN. We find that models with ReLU MLP and SIREN achieves similar quantitative performance. And qualitatively, LIIF with SIREN has sharper boundaries than that with RELU MLP. We also conclude that the bottleneck of the super-resolution problem lies in the design of feature encoder, and not in the design of local implicit neural representations.

1. Introduction

The motivation of single image super-resolution is to convert a low-resolution (LR) image into a high-resolution (HR) image. Single image super-resolution (SISR) is quite different from multi-image super-resolution (MISR) because the input to SISR is one single LR image while the inputs to MISR are multiple LR images. SISR are generally considered more difficult than MISR since we will have much less information if we only have one single image.

Recently, deep learning has made tremendous progress in the area of SISR with the help of emerging high-performance GPUs. Researchers have also proven the success of deep learning on SISR via various neural network archtectures. Traditional deep learning approach for SISR, such as enhanced deep residual network for single image super-resolution (EDSR) [1] and residual dense network (RDN) [2], propose to use convolutional residual blocks for the feature encoders and dedicated convolutional layers for upscaling modules. This kind of CNN-based upscaling module can only scale the image by a fixed factor. Emerging methods, such as local implicit image function (LIIF) propose to replace these upscaling modules with an implicit neural representation. An implicit neural representation can be regard as a continuous function parameterized by a simple multilayer perception (MLP). Since this function is continuous, LIIF is able to represent the image at any resolution given the coordinates. Researchers have also made huge progresses in the area of implicit neural representations. Sinusoidal representation networks (SIREN) [3], which replace the ReLU activation functions inside MLP with the sinusoidal activation functions, tend to take less epochs for training and generalize well with low dimensional inputs.

Table 1: Comparison between related works

Methods	CNN Feature Encoder	Fixed Upscaling Module	Genralized MLP
RDN [2]	✓	✓	✗
EDSR [1]	✓	✓	✗
LIIF [4]	✓	✗	✓
LIIF+SIREN [3] (ours)	✓	✗	✓

1.1. Paper Contributions and Organization

In this paper, we propose to compare and contrast the performances of LIIF [4] under two different activation functions, ReLU and Sine activation functions. Our specific contribution includes:

- We propose to replace the regular ReLU MLP inside LIIF with SIREN.
- Quantitatively, we find that models with ReLU and Sine activation functions achieve basically the same performance under the same feature encoders.
- Qualitatively, LIIF with Sine activation functions has sharper boundaries while LIIF with ReLU activation functions has obscure boundaries.
- We also conclude that the bottleneck of the super-resolution problem lies in the design of feature encoder and not in the design of local implicit neural representations.

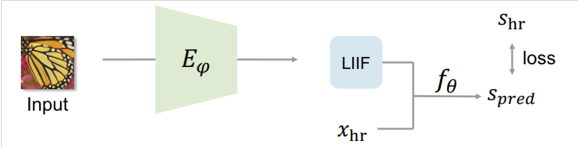


Figure 1: Overall neural network architecture [4]

Section 2 talks about related works in SISR and implicit image representations. Section 3 briefly talks about the neural network architecture for feature encoders and local implicit image function. Section 4 describes the experimental setup and both quantitative and qualitative results. Section 5 analyzes the bottleneck of LIIF by comparing different models. Section 6 discusses the limitation and possible future works. Finally, Section 7 concludes the paper.

2. Related Works

Super-resolution Table 1 clearly demonstrates the differences between currently existing deep learning methods for super-resolution. Prior to the existence of local implicit image functions [4], traditional deep learning methods for super-resolution, such as RDN [2] and EDSR [1], rely on a fixed CNN upscaling module. The upscaling module aims to scale the LR image by a fixed factor. Thus, when people want to scale the image by a different factor, they need to train the entire upscaling module again. Recently, LIIF proposes to replace the fixed CNN upscaling module with a generalized MLP. In other words, LIIF uses an implicit neural representation (MLP) to represent the image. Since the implicit neural representation is continuous, LIIF is able to scale the LR image by an arbitrary factor.

Implicit neural representations SIREN [3] proposes to replaces the ReLU activation functions with Sine activation functions. They can represent the 2-D images, the corresponding gradients and Laplacians way better than the regular ReLU MLP.

3. Approaches

This section presents the overall neural network architecture. The entire model can be divided into two parts: convolutional neural network (CNN) based feature encoder and multilayer perceptron (MLP) based local implicit image function (LIIF). As shown in Figure 1, the input to the CNN-based feature encoder, E_φ, is a LR image. Then, the feature encoder, E_φ, outputs a feature mapping, $M^{(i)} \in \mathbb{R}^{H\times W\times D}$ for every input LR image, $I^{(i)} \in \mathbb{R}^{H\times W}$, where H is the height of the image, W is the width of the image and D is the number of features. Furthermore, some particular latent codes in the feature mapping, $M^{(i)}$, together with the 2-D coordinate of the high resolution image, x_{hr} are fed into a MLP, f_θ, which is parameterized by θ. The output of this MLP is the corresponding predicted RGB value, s_{pred}. This MLP is called local implicit image function. Finally, L_1 loss is computed betweern the predicted RGB value, $s_{predict}$ and the ground truth, s_{hr}.

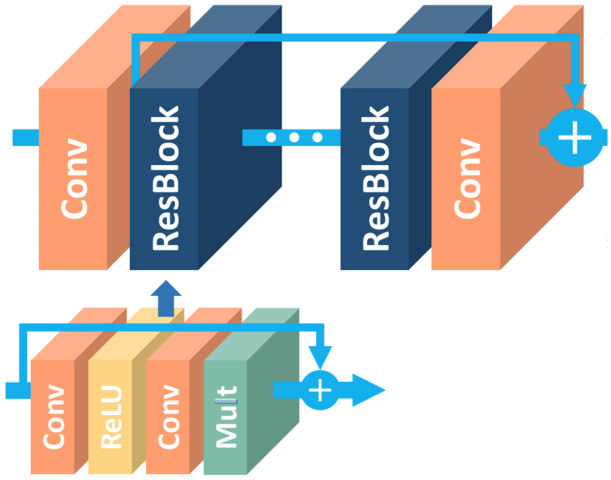


Figure 2: CNN-based residual network [1]

3.1. Feature Encoder

Like [4], we use a CNN-based residual network as our feature encoder, E_φ. Specifically, we use the same feature encoder as EDSR [1]. Each residual block inside this feature encoder is made of two convolutional layers, one ReLU activation layer and one multiplication layer. In this paper, we can omit the multiplication layer since we always set the multiplication scale as 1 (Figure 2).

3.2. Local Implicit Image Function

[4] proposes to use a local implicit image function (LIIF), f_θ as a continuous representation of an image, $I^{(i)}$. For the image $I^{(i)}$, the RGB pixel value at coordinate x_q is formulated as follows:

$$I^{(i)}(x_q) = f_\theta(z^*, x_q - v^*), \tag{1}$$

where z^* is the nearest latent code from x_q in feature mapping $M^{(i)}$ and v^* is the 2-D coordinates of the nearest latent code.

In order to increase the amount of information in the feature mapping $M^{(i)}$, LIIF simply concatenates a 3×3 neighborhood in the original feature mapping $M^{(i)}$ to form a new feature mapping $\hat{M}^{(i)}$. Each new latent code $\hat{M}^{(i)}_{j,k}$ can be formulated as follows:

$$\hat{M}^{(i)}_{j,k} = \text{Concat}(\{M^{(i)}_{j+l,k+m}\}_{l,m\in\{-1,0,1\}}) \tag{2}$$

In order to ensure smooth transitions between nearest latent code, LIIF ensembles the predictions obtained from a

2×2 latent code neighborhood via weighted sum. Each prediction is weighted by their corresponding area of rectangle.

Finally, LIIF adds the width and height of the query pixels to the inputs of the MLP.

4. Experiments

In this section, we will go over the experimental setups and both quantitative and qualitative results.

4.1. Datasets

We train all our models on the DIV2K datasets [5]. The DIV2K datasets contains totally 1000 2K-resolution images, where 800 images are in training set, 100 images are in validation set, and 100 images are in testing set. Then, we test our models on both the DIV2K dataset and benchmark dataset. The benchmark datasets are comprised of 4 classical datasets: Set5 [6], Set14 [7], B100 [8] and Urban100 [9].

4.2. Model Details

Table 2: Feature encoder specifications

Names	# Params	# ResBlocks	# Features, D
EDSR	1.5M	16	64
EDSR-LESS	579.3K	16	32

We have totally 5 different models, EDSR-LIIF-ReLU (baseline), EDSR-LIIF-SIREN-30, EDSR-LIIF-SIREN-15, EDSR-LESS-LIIF-ReLU (baseline) and EDSR-LESS-LIIF-SIREN-30. We define our baseline models as models with regular ReLU activation functions. These 5 models can be further divided into two groups. One group uses EDSR feature encoder, and the other group uses EDSR-LESS feature encoder. As shown in Table 2, EDSR-LESS feature encoder has significantly less parameters (around one third) than EDSR feature encoder. The local implicit image functions with ReLU MLP and SIREN have the same number of hidden layers (5) and the same hidden size (256). SIREN-30 has slightly bigger weight initialization than SIREN-15. We would like to experiment with different weight initializations for SIREN because different weight initializations lead to different expressive power of the networks [3].

4.3. Training Details

We are training our models on the Google compute platform (GCP), using a single Nvidia V100 GPU. Each model is trained for 600 epochs. As shown in Figure 3, the learning rates of the ADAM optimizer start with 1×10^{-4} for all the models. Then, the learning rates are halved every

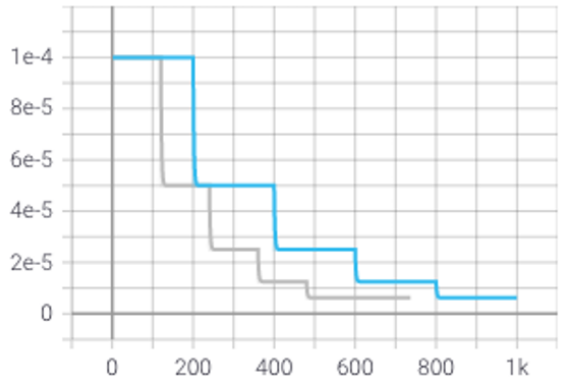


Figure 3: Learning rate vs. epochs (grey: SIREN, blue: ReLU MLP)

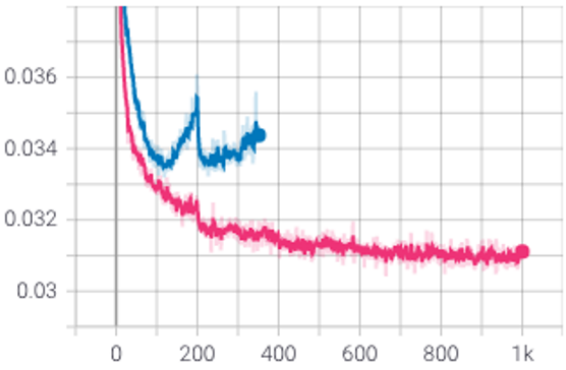


Figure 4: Training loss vs. epochs when lr is halved every 200 epochs (blue: SIREN, red: ReLU MLP)

120 epochs for the models with SIREN while the learning rates are halved every 200 epochs for the models with ReLU MLP. We choose to halve the learning rates for models with SIREN every 120 epochs because model starts to overfit at epoch 120. As shown in Figure 4, the training loss for the model with SIREN starts to increase at epoch 120 when the learning rate is halved every 200 epochs.

4.4. Quantitative Results

Table 3 shows the performances of all five models on the benchmark datasets. And Table 4 demonstrates the performances of all five models on the DIV2K datasets [5]. Under the same feature encoder, we find that models with ReLU and Sine activation functions achieve similar performance. We used to believe that the bottleneck of this super-resolution problem is the design of the implicit neural representation. Now, we find that the actual bottleneck lies in the design of feature encoder.

Table 3: Performance comparisons on the benchmark datasets (PSNR / dB).

Datasets	Methods	In-domain			Out-of-domain	
		x2	x3	x4	x6	x8
Set5	EDSR-LIIF-ReLU (baseline)	37.96	34.40	32.11	28.85	26.95
	EDSR-LIIF-SIREN-30	37.96	34.33	32.11	28.88	26.94
	EDSR-LIIF-SIREN-15	37.96	34.36	32.13	28.85	26.93
	EDSR-LESS-LIIF-ReLU (baseline)	37.79	34.17	32.03	28.72	26.79
	EDSR-LESS-LIIF-SIREN-30	37.76	34.15	31.99	28.66	26.66
Set14	EDSR-LIIF-ReLU (baseline)	33.60	30.32	28.59	26.45	24.93
	EDSR-LIIF-SIREN-30	33.62	30.31	28.56	26.41	24.91
	EDSR-LIIF-SIREN-15	33.61	30.32	28.59	26.44	24.90
	EDSR-LESS-LIIF-ReLU (baseline)	33.42	30.19	28.44	26.29	24.80
	EDSR-LESS-LIIF-SIREN-30	33.38	30.14	28.43	26.27	24.77
B100	EDSR-LIIF-ReLU (baseline)	32.14	29.09	27.59	25.84	24.78
	EDSR-LIIF-SIREN-30	32.15	29.08	27.57	25.83	24.78
	EDSR-LIIF-SIREN-15	32.15	29.09	27.58	25.83	24.78
	EDSR-LESS-LIIF-ReLU (baseline)	32.04	28.98	27.48	25.74	24.71
	EDSR-LESS-LIIF-SIREN-30	32.03	28.96	27.46	25.73	24.69
Urban100	EDSR-LIIF (baseline)	32.02	28.15	26.08	23.76	22.43
	EDSR-LIIF-SIREN-30	32.04	28.13	26.04	23.71	22.41
	EDSR-LIIF-SIREN-15	32.01	28.15	26.07	23.74	22.43
	EDSR-LESS-LIIF-ReLU (baseline)	31.57	27.78	25.76	23.49	22.23
	EDSR-LESS-LIIF-SIREN-30	31.52	27.72	25.72	23.47	22.19

Table 4: Performance comparisons on the DIV2K datasets (PSNR / dB).

Methods	In-domain			Out-of-domain				
	x2	x3	x4	x6	x12	x18	x24	x30
EDSR-LIIF-ReLU (baseline)	34.62	30.93	28.98	26.74	23.70	22.17	21.18	20.49
EDSR-LIIF-SIREN-30	34.60	30.90	28.94	26.70	23.67	22.15	21.16	20.47
EDSR-LIIF-SIREN-15	34.61	30.92	28.96	26.71	23.69	22.16	21.18	20.48
EDSR-LESS-LIIF-ReLU (baseline)	34.39	30.72	28.80	26.57	23.58	22.07	21.09	20.41
EDSR-LESS-LIIF-SIREN-30	34.37	30.69	28.77	26.54	23.55	22.04	21.08	20.40

4.5. Qualitative Results

Figure 5 shows the qualitative performance on an alphabetic image from the Set14 testing set. The LR input image is cropped from the original image directly without any downsampling and fed into two models EDSR-LIIF-ReLU and EDSR-LIIF-SIREN-15. The image produced by the model with SIREN has slightly sharper edges around the character "w" in red circle (Figure 5). Figure 6 shows the qualitative performance on a natural image from the DIV2K testing set. The LR input image is bicubic downsampled by a factor of 30 from the original image and fed into two models EDSR-LIIF-ReLU and EDSR-LIIF-SIREN-15. The image produced by the model with SIREN has slightly sharper boundries on the chest between the red color and the white color in yellow circle (Figure 6).

5. Analysis and Evaluation

5.1. Bottleneck

The actual bottleneck of this super-resolution method lies in the design of feature encoder. According to Table 3 and Table 4, the models with smaller feature encoders (EDSR-LESS) tend to perform worse than that with bigger feature encoders (EDSR). What's more, the performance of each model are almost the same quantitatively under same feature encoders although we can observe slightly different edge representations.

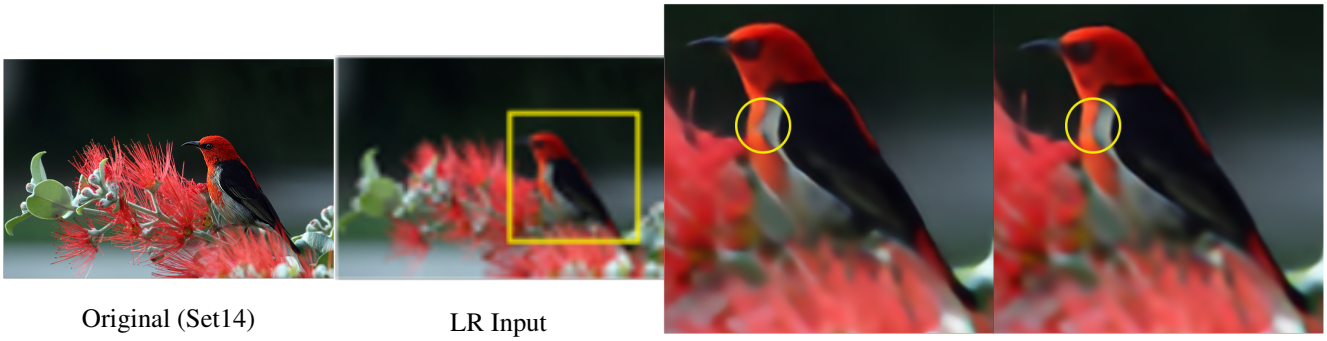
6. Discussion

6.1. Limitation

Deep learning for super-resolution are susceptible to adversarial attacks like any other general deep learning appli-

Original (Set14) LR Input EDSR-LIIF-ReLU (baseline) EDSR-LIIF-SIREN-15 (ours)

Figure 5: Qualitative Performance on Set14 testing set ($\times$ 30)



Original (Set14) LR Input EDSR-LIIF-ReLU (baseline) EDSR-LIIF-SIREN-15 (ours)

Figure 6: Qualitative Performance on DIV2K testing set ($\times$ 30)

cations. People should be extra careful if they would like to apply these kinds of methods on medical imaging and security surveillance. There are no theoretical guarantees on the results produced by deep learning. Thus, if some medical personnel would like to scale their patients' MRI images by a large scale, they will have to understand the resultant HR MRI images might not be true representations. Similarly, the photos from security cameras scaled by deep learning shall never be used as evidences on the court.

6.2. Future Work

Compared with digital single lens reflex (DSLR) cameras, reflex 4D plenoptic cameras, or light-field cameras can only capture row-resolution images due to their inherent design. DSLR cameras capture a single photo at a time while light-field cameras capture more than one photo, possibly over 200 photos from different angles, at one time. Given the same camera sensor, the resolutions of photos taken by light-field cameras are inevitably lower than those taken by DSLR cameras. Traditional super-resolution methods, such as EDSR and RDN, can only scale the LR images within the domain of training sets. Images in light-field camera data sets tend to have lower resolutions. Thus, these methods are not desirable for light-field camera images.

With the help of LIIF, we can scale the LR light-field images by an arbitrary scale. And we may just use 4-D coordinates as the inputs to the plenoptic implicit image functions. The extra 2 dimensions represents the positions of different camera angles.

7. Conclusion

In this paper, we successfully investigate the performances of LIIF under two different activation functions, Sine and ReLU activation functions. Models with both activation functions have similar performances quantitatively. However, the models with Sine activation functions have slightly sharper edges than those with ReLU activation functions qualitatively. We also conclude that the bottleneck of the super-resolution problem lies in the design

of feature encoder, and not in the design of local implicit neural representations.

8. Acknowledgement

We would like to thank Professor Gordon Wetzstein and Alex Bergman for their insightful discussion and help.

References

[1] B. Lim, S. Son, H. Kim, S. Nah, and K. Mu Lee, "Enhanced deep residual networks for single image super-resolution," in *Proceedings of the IEEE conference on computer vision and pattern recognition workshops*, pp. 136–144, 2017.

[2] Y. Zhang, Y. Tian, Y. Kong, B. Zhong, and Y. Fu, "Residual dense network for image super-resolution," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 2472–2481, 2018.

[3] V. Sitzmann, J. N. Martel, A. W. Bergman, D. B. Lindell, and G. Wetzstein, "Implicit neural representations with periodic activation functions," in *Proc. NeurIPS*, 2020.

[4] Y. Chen, S. Liu, and X. Wang, "Learning continuous image representation with local implicit image function," *arXiv preprint arXiv:2012.09161*, 2020.

[5] R. Timofte, E. Agustsson, L. Van Gool, M.-H. Yang, and L. Zhang, "Ntire 2017 challenge on single image super-resolution: Methods and results," in *Proceedings of the IEEE conference on computer vision and pattern recognition workshops*, pp. 114–125, 2017.

[6] M. Bevilacqua, A. Roumy, C. Guillemot, and M. L. Alberi-Morel, "Low-complexity single-image super-resolution based on nonnegative neighbor embedding," 2012.

[7] R. Zeyde, M. Elad, and M. Protter, "On single image scale-up using sparse-representations," in *International conference on curves and surfaces*, pp. 711–730, Springer, 2010.

[8] D. Martin, C. Fowlkes, D. Tal, and J. Malik, "A database of human segmented natural images and its application to evaluating segmentation algorithms and measuring ecological statistics," in *Proceedings Eighth IEEE International Conference on Computer Vision. ICCV 2001*, vol. 2, pp. 416–423, IEEE, 2001.

[9] J.-B. Huang, A. Singh, and N. Ahuja, "Single image super-resolution from transformed self-exemplars," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 5197–5206, 2015.