B-spline Texture Coefficients Estimator for Screen Content Image Super-Resolution

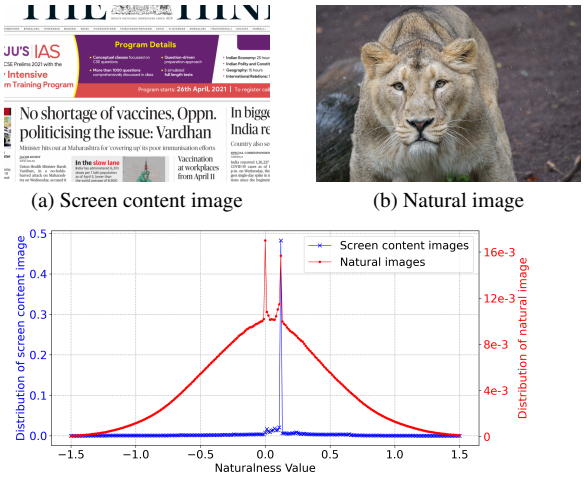
Byeonghyun Pak* Jaewon Lee* Kyong Hwan Jin†
Daegu Gyeongbuk Institute of Science and Technology (DGIST), Korea
{w9bkje, ljw3136, kyong.jin}@dgist.ac.kr

Abstract

Screen content images (SCIs) include many informative components, e.g.*, texts and graphics. Such content creates sharp edges or homogeneous areas, making a pixel distribution of SCI different from the natural image. Therefore, we need to properly handle the edges and textures to minimize information distortion of the contents when a display device's resolution differs from SCIs. To achieve this goal, we propose an implicit neural representation using B-splines for screen content image super-resolution (SCI SR) with arbitrary scales. Our method extracts scaling, translating, and smoothing parameters of B-splines. The followed multi-layer perceptron (MLP) uses the estimated B-splines to recover high-resolution SCI. Our network outperforms both a transformer-based reconstruction and an implicit Fourier representation method in almost upscaling factor, thanks to the positive constraint and compact support of the B-spline basis. Moreover, our SR results are recognized as correct text letters with the highest confidence by a pre-trained scene text recognition network. Source code is available at* https://github.com/ByeongHyunPak/btc.

1. Introduction

With the rapid development of multimedia applications, screen content images (SCIs) have been common in people's daily life. Many users interact with SCIs through various display terminals, so resolution mismatch between a display device and SCIs occurs frequently. In this regard, we need to consider a flexible reconstruction at arbitrary magnification from low-resolution (LR) SCI to its high-resolution (HR). As in Figs. 1a and 1b, SCI has discontinuous tone contents, whereas natural image (NI) has smooth and continuous textures. Such characteristics are observed as a Gaussian distribution in the naturalness value of NIs [25] and sharp fluctuations in the naturalness value of SCIs in Fig. 1c. This observation leads to a screen content image super-resolution (SCI SR) method considering such distributional properties. However, most SR methods [4, 5, 8–10, 28, 29] are applied to NIs.



(a) Screen content image (b) Natural image

(c) Naturalness value distributions for 500 images per each class

Figure 1. **Comparison on naturalness value distribution [25] between screen content images and natural images.**

Recently, Yang *et al*. proposed a novel SCI SR method based on a transformer, implicit transformer super-resolution network (ITSRN) [26]. Since ITSRN evaluates each pixel value by a point-to-point implicit function through a transformer architecture, it outperforms CNN-based methods [28, 29]. However, even though ITSRN represents SCI's characters (*e.g*., sharp edges or homogeneous areas) continuously, it has a large model size leading to inefficient memory consumption and slow inference time.

Meanwhile, Chen *et al*. first introduced implicit neural representation (INR) to single image super-resolution (SISR) [4]. The implicit neural function enables arbitrary scale super-resolution by jointly combining the continuous query points and the encoded feature of the input LR image. Nevertheless, such implicit neural function, implemented with a multi-layer perceptron (MLP), is biased to learn the low-frequency components, called spectral bias [15].

*Equal contribution.
†Corresponding author.

Lee and Jin suggested the local texture estimator (LTE) upon INR to overcome the above problem [8]. LTE estimates the frequencies and corresponding amplitude features from the input LR image and feeds them into an MLP with the Fourier representation. Here, projecting input into a high-dimensional space with the sinusoids in Fourier representation allows the implicit neural function to learn high-frequency details. However, since LTE expresses a signal with a finite sum of sinusoids, it has a risk for the reconstructed values to under/overshoot at the discontinuities of SCIs, called the Gibbs phenomenon. This phenomenon often produces incorrect information about SCIs. Thus, we need to restore HR SCIs with fewer parameters, fewer computation costs, and less distortion of contents.

In this paper, we propose a B-spline Texture Coefficients estimator (BTC) utilizing INR to represent SCIs continuously. BTC predicts scaling (*coefficients*), translating (*knots*), and smoothing (*dilations*) parameters of B-splines from the LR image. Then, inspired by Lee and Jin [8], we project the query point's coordinate into the high-dimensional space with 2D B-spline representations and feed them into MLP. Since the B-spline basis has a positive constraint and compact support, BTC preserves discontinuities well without under/overshooting.

Our main contributions are: (I) We propose a B-spline Texture Coefficients estimator (BTC), which estimates B-spline features (*i.e.*, coefficients, knots, and dilations) for SCI SR. (II) With a 2D B-spline representation, we achieve better performances with fewer parameters and less memory consumption. (III) We demonstrate that B-spline representation is robust to over/undershooting aliasing when reconstructing HR SCIs, owing to positive constraint and compact support of the B-spline basis function.

2. Related works

2.1. Screen content image super-resolution

Screen content images (SCIs) are composed of computer-rendered content, *e.g.*, graphics and texts. Therefore, as shown in Fig. 1c, the pixel distribution of screen content images differs from that of natural images [25]. Paying attention to this point, Yang *et al*. tackle the SR problems in SCIs when the limited communications bandwidth cannot send high-resolution content [26]. The authors utilized a transformer-based algorithm, ITSRN, to infer pixel values using relative coordinates between low-resolution and high-resolution SCIs. In addition, they applied an implicit position encoding to aggregate neighboring pixel values to represent SR images within the continuous regime. However, ITSRN's large model size requires more memory and inference time. In contrast, our proposed algorithm represents edge-highlighted features even with fewer training parameters and computational costs.

2.2. Implicit neural representation (INR)

Recent implicit neural representations (INR) have achieved a good ability to represent signals implicitly and continuously from partial observations [4, 12, 17]. The implicit neural function, implemented by multi-layer perceptron (MLP), takes coordinates as input and returns the corresponding signal values. However, a combination of an MLP and ReLU activation suffers from spectral bias [15], a limitation in representing high-frequency details. Sitzmann *et al*. substitute a ReLU with a sinusoidal activation function to overcome the spectral bias [17]. In addition, Tancik *et al*. [18] and Mildenhall *et al*. [12] project the input coordinates to a high-dimensional space with a set of sinusoids (position encoding). With the same purpose, Lee and Jin [8] proposed a local texture estimator (LTE) for single image super-resolution to estimate dominant frequencies and corresponding amplitudes with Fourier representation. However, the finite sum of sinusoids in Fourier representation can make the restored signal under/overshoot at the discontinuities of the signal, Gibbs phenomenon. This paper utilizes B-spline basis functions to reconstruct SCIs with high discontinuity better.

2.3. B-spline representation

B-spline has gained the attention of signal processing society in terms of stable and good signal representations [16, 21, 22]. Splines, including a B-spline basis, have been extensively studied in signal representations [19, 20]. Recently, a uniform B-spline basis with trainable coefficients was used as a position encoder of given Cartesian coordinates. This leads to a better representation of a 3D signal than frequency encoding [23]. Prasad *et al*. [14] proposed a trainable non-uniform rational B-spline (NURBS) layer to fit a surface from point clouds. However, the mentioned works [14,23] are not demonstrated in SISR. This paper utilizes a non-uniform 2D B-spline representation suitable for reconstructing SCIs because of its compactness and positivity constraints. Our method estimates not only coefficient information but also knots and dilations from local regions of the input image and upscale factor.

3. Problem formulation

We tackle single image super-resolution (SISR) problems for screen content images. To prevent misleading information by asymmetric scaling, SISR for screen content images considers a unified aspect ratio (r:r), *i.e.*, the same upscaling ratio between horizontal and vertical axes.

Our goal is to reconstruct a high-resolution RGB image $\mathbf{I_{HR}} \in \mathbb{R}^{3\times rH\times rW}$ from a low-resolution RGB image $\mathbf{I_{LR}} \in \mathbb{R}^{3\times H\times W}$. To acquire $\mathbf{I_{LR}}$, we utilize the bicubic interpolation degradation model from $\mathbf{I_{HR}}$ to $\mathbf{I_{LR}}$ given by

$$\overrightarrow{\mathbf{I}}_{\mathbf{LR}} = \mathbf{D}_r \mathcal{T}_{k_b} \overrightarrow{\mathbf{I}}_{\mathbf{HR}}, \tag{1}$$

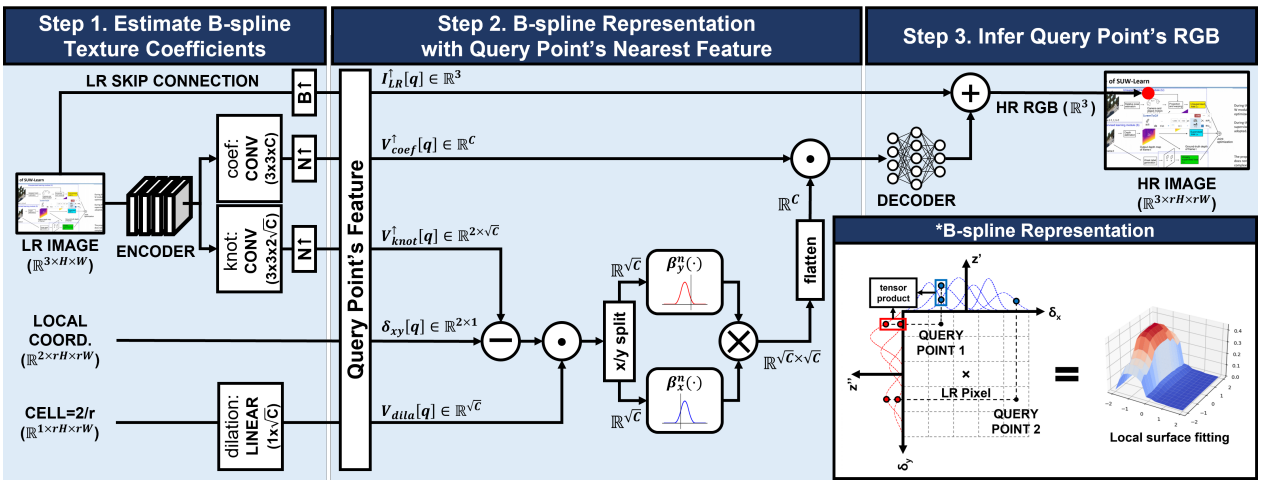


Figure 2. **Overall pipeline of our B-spline Texture Coefficients estimator (BTC).** $\oplus, \ominus, \odot, \otimes$, B↑ and N↑ denote element-wise addition, element-wise subtraction, element-wise multiplication, tensor product, Bilinear up and Nearest-Neighborhood up, correspondingly.

where k_b is a cubic convolutional kernel, $\mathcal{T}$ is a block Toeplitz matrix from 2D convolution, and $\mathbf{D}_r$ is a downsampling operator with a factor of r. To represent bicubic-interpolated signals in the implicit neural representation, we use non-uniform B-spline basis ($\beta^n : \mathbb{R} \mapsto \mathbb{R}$) with the generalized form [21, 23] :

$$s(x) = \sum_{t\in\mathcal{I}} c[t]\beta^n\left(\frac{x-k[t]}{d[t]}\right), \tag{2}$$

where $c[t]$ is the $t-$th coefficient, $k[t]$ is the $t-$th continuous knot , $d[t]$ is the $t-$th continuous dilation variable, and $\mathcal{I}$ is a index set of knots. The definitions of $\beta^0(\cdot)$ and $\beta^3(\cdot)$ are given in Sec. 4.2. From now on, we explain the concept of implicit neural representation with a B-spline basis to represent 2D signals.

Implicit Neural Representation A decoder f_θ, implemented with a trainable MLP, θ, maps both latent codes and local coordinates into query point's RGB values; $f_\theta(\mathbf{z}, \mathbf{x}) : (\mathcal{Z}, \mathcal{X}) \mapsto \mathcal{Q}$. Here, $\mathbf{z} \in \mathcal{Z}$ is a latent variable from the encoder E_φ, $\mathbf{x} \in \mathcal{X}$ is a continuous 2D coordinate of $\mathbf{I_{HR}}$, and $\mathcal{Q} \in \mathbb{R}^3$ is an predicted RGB space from f_θ. The latent code $\mathbf{z} \in \mathbb{R}^{D\times H\times W}$ has the same width and height with $\mathbf{I_{LR}}$. Therefore, the query point's RGB values ($\mathbf{I_{HR}}(\mathbf{x}) \in \mathbb{R}^3$) at a coordinate $\mathbf{x} \in \mathbb{R}^2$ are calculated as

$$\mathbf{q}(\mathbf{I_{LR}}, \mathbf{x}; \Theta) = \sum_{t\in\mathcal{N}} w_t f_\theta(\mathbf{z}_t, \mathbf{x}-\mathbf{x}_t, s), \tag{3}$$

$\mathbf{z} = E_\varphi(\mathbf{I_{LR}})$, $\Theta = \{\theta; \varphi\}$, $\mathcal{N} \in \mathbb{Z}^4$ is a set of indices for four nearest latent codes around $\mathbf{x}$, w_t is the bilinear interpolation weight corresponding to the latent code t (referred to as the local ensemble weight [4]), $\mathbf{z}_t \in \mathbb{R}^D$ is the nearest latent feature vector from $\mathbf{x}$, $\mathbf{x}_t \in \mathbb{R}^2$ is the coordinate of the latent code t, and s is a cell value represented with an upscaling factor [4]. With a series of M query points from N images such as $(\mathbf{x}_m, \mathbf{I}^n_{\mathbf{HR}}(\mathbf{x}_m))$, $m = 1, \ldots, M$ and $n = 1, \ldots, N$, the learning problem is defined as

$$\hat{\Theta} = \arg\min_{\Theta} \sum_{m,n}^{M,N} \|\mathbf{I}^n_{\mathbf{HR}}(\mathbf{x}_m) - \mathbf{q}(\mathbf{I}^n_{\mathbf{LR}}, \mathbf{x}_m; \Theta)\|_1. \tag{4}$$

Learning non-uniform bivariate B-splines Local Texture Estimator (LTE) [8], which predicts essential Fourier features, was proposed for natural images to resolve the spectral bias of an implicit neural function. To exploit both position encoding [12] and Fourier feature mapping [18], LTE embeds an input into the Fourier space before an MLP. However, screen content images are synthetic and rendered by computer software, so Fourier features do not sufficiently represent SCIs. Thus, we use non-uniform B-splines (Eq. (2)) for feature embedding. We call our algorithm BTC. When we add another implicit representation with BTC inside f_θ, the local implicit neural representation in Eq. (3) can be written as follows:

$$\mathbf{q}(\mathbf{I_{LR}}, \mathbf{x}; \Theta, \psi) = \sum_{t\in\mathcal{N}} w_t f_\theta(g_\psi(\mathbf{z}_t, \mathbf{x}-\mathbf{x}_t, s)) \tag{5}$$

where s is a cell value which means the size of the query pixel ($= 2/r$) and $g_\psi(\cdot)$ denotes the BTC. BTC ($g_\psi(\cdot)$) consists of three estimators; (1) a coefficient estimator ($g_c : \mathbb{R}^D \mapsto \mathbb{R}^C$), (2) a knot estimator ($g_k : \mathbb{R}^D \mapsto \mathbb{R}^{2\sqrt{C}}$), and (3) a dilation estimator ($g_d : \mathbb{R}^1 \mapsto \mathbb{R}^{\sqrt{C}}$). Given a local-grid coordinate $\boldsymbol{\delta}_t (= \mathbf{x} - \mathbf{x}_t) \in \mathbb{R}^2$ and cell value $s (= 2/r) \in \mathbb{R}^1$, the encoding function $g_\psi : (\mathbb{R}^D, \mathbb{R}^2, \mathbb{R}^1) \mapsto \mathbb{R}^C$ is defined as

$$g_\psi(\mathbf{z}_t, \boldsymbol{\delta}_t, s) = \mathbf{c}_t \odot vec\left[\beta^n\left(\frac{\boldsymbol{\delta}^y_t - \mathbf{k}^y_t}{\mathbf{d}}\right) \otimes \beta^n\left(\frac{\boldsymbol{\delta}^x_t - \mathbf{k}^x_t}{\mathbf{d}}\right)^T\right], \tag{6}$$

$$\text{where } \mathbf{c}_t = g_c(\mathbf{z}_t),\ [\mathbf{k}^x_t, \mathbf{k}^y_t] = g_k(\mathbf{z}_t),\ \mathbf{d} = g_d(s). \tag{7}$$

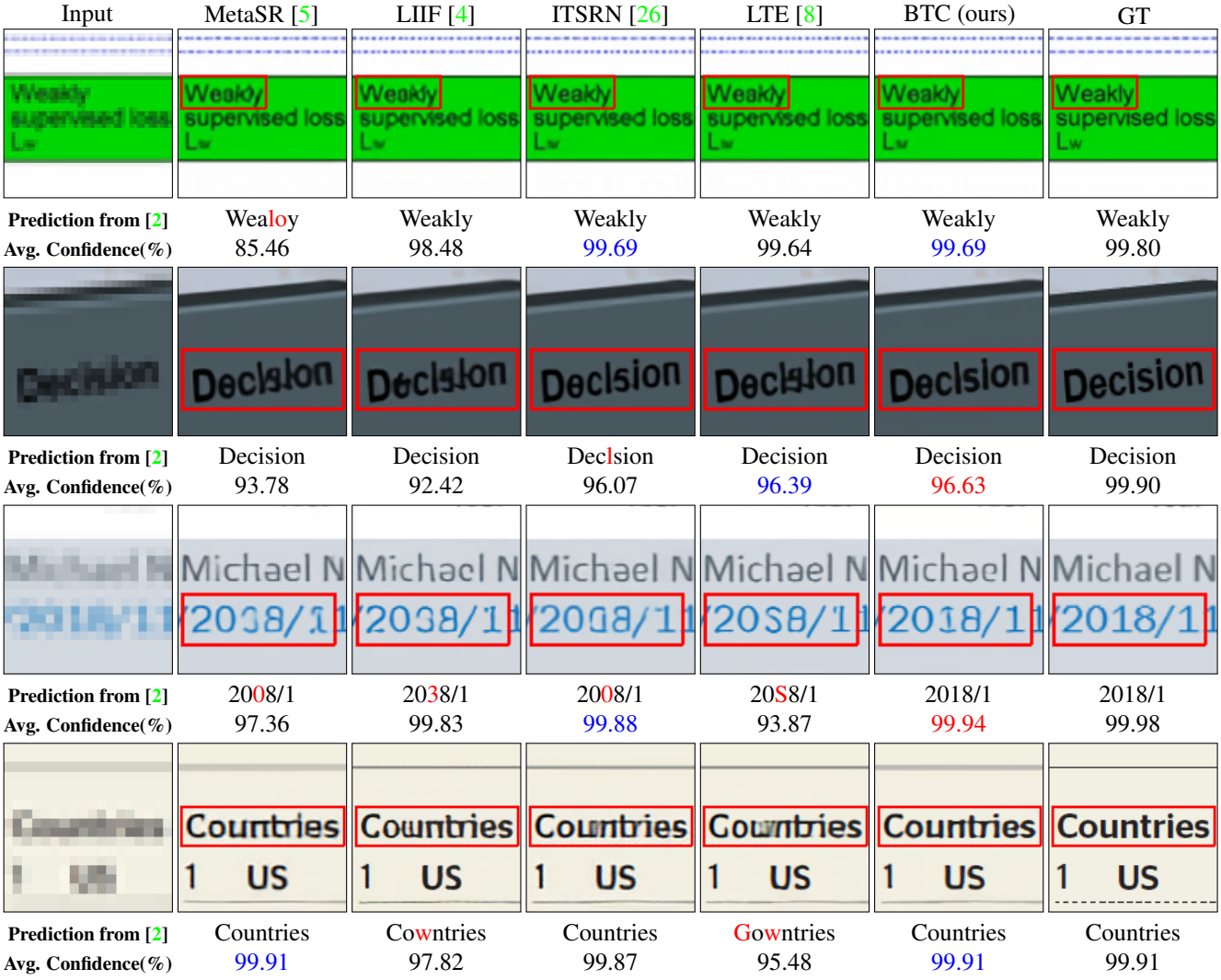


Figure 3. **Qualitative comparison with other arbitrary-scale SR methods** [4, 5, 8, 26] for $\times 3$ (first), $\times 4$ (second), $\times 5$ (third), and $\times 7$ (last). All methods are trained on SCI1K train set with RDN [29] encoder. We report the scene text recognition (STR) results of the red-highlighted boxes using a pre-trained STR network [2], *i.e.*, prediction for the red box region and its average confidence. The best and second confidences are in red and blue, respectively.

$\mathbf{c}_t \in \mathbb{R}^C$ is a coefficient vector for a latent code $\mathbf{z}_t$, $[\mathbf{k}_t^x, \mathbf{k}_t^y] \in \mathbb{R}^{2\sqrt{C}}$ indicate predicted knots from a latent code $\mathbf{z}_t$, $\mathbf{d} \in \mathbb{R}^{\sqrt{C}}$ denote dilation variables, β^n is a $n-$th order B-spline ($\beta^n : \mathbb{R}^{\sqrt{C}} \mapsto \mathbb{R}^{\sqrt{C}}$) , and $\otimes$ represents tensor product (outer product between two vectors). Since MLPs with ReLUs are incompetent at extrapolate unseen non-linear space [24], we use $s = max(s, s_{tr})$, where s_{tr} indicates the minimum cell size during training.

Evaluated bivariate B-splines ($g_\psi(\cdot)$) for a local area $r \times r$ for a query point $(\boldsymbol{\delta}_x, \boldsymbol{\delta}_y)$ is represented with multiplications between B-spline elements given by:

$$g_\psi(\mathbf{z}_t, \boldsymbol{\delta}_t, s)[q] = \mathbf{c}_t[q]\, \beta^n \left(\frac{\boldsymbol{\delta}_t^y - \mathbf{k}_t^y[i]}{\mathbf{d}[i]} \right) \beta^n \left(\frac{\boldsymbol{\delta}_t^x - \mathbf{k}_t^x[j]}{\mathbf{d}[j]} \right), \tag{8}$$

where $q = i \cdot \sqrt{C} + j$ and $i, j = 0, \ldots, \sqrt{C} - 1$. From this description, we notice that a $r \times r$ local area is fitted with C different B-splines consisting of estimated dilations $\mathbf{d}$ and knots $\mathbf{k}$. In practice, we use a third-order B-spline ($n = 3$). Because B-spline is compactly supported and positive in all ranges [21], it sufficiently represents synthetic signals that frequently have sharp edges with less undershooting or overshooting. Contrarily, LTE [8] is inconsistent with synthetic signals due to its Fourier basis. This is explained in the discussion.

As in [8], a long skip connection improves graphical textures in residuals and stabilizes convergence. Our algorithm is formulated as follows:

$$\hat{\mathbf{q}}(\mathbf{x}) = \mathbf{q}(\mathbf{I}_{\mathbf{LR}}, \mathbf{x}; \Theta, \psi) + \mathbf{I}_{\mathbf{LR}}^{\uparrow}(\mathbf{x}) \tag{9}$$

Train set: SCI1K (n=800)			In-training-scale			Out-of-training-scale					
Test set	**Method**	**# Params.**	**×2**	**×3**	**×4**	**×5**	**×6**	**×7**	**×8**	**×9**	**×10**
SCI1K (n = 200)	Bicubic	-	28.81	25.15	23.18	22.02	21.23	20.72	20.26	19.96	19.67
	RDN [29]	21.97M	38.45	33.59	29.81	-	-	-	-	-	-
	MetaSR [5]	22.42M	38.57	33.67	30.12	27.52	26.13	23.91	23.19	22.02	21.73
	LIIF [4]	22.32M	38.65	33.97	30.55	27.77	26.07	23.99	23.24	22.18	21.81
	ITSRN [26]	22.62M	38.74	34.32	30.82	28.15	26.07	24.36	23.12	22.36	21.77
	LTE [8]	22.53M	39.14	34.50	30.93	28.22	26.19	24.28	23.17	22.39	21.85
	BTC (ours)	22.40M	39.17	34.58	31.10	28.33	26.31	24.47	23.38	22.48	21.89
SCID (n = 40)	Bicubic	-	25.22	22.78	21.60	20.90	20.42	20.04	19.77	19.51	19.29
	RDN [29]	21.97M	34.00	28.34	25.74	-	-	-	-	-	-
	MetaSR [5]	22.42M	33.84	29.08	25.76	23.62	22.38	21.59	21.07	20.71	20.41
	LIIF [4]	22.32M	34.24	29.10	25.89	23.77	22.53	21.73	21.21	20.84	20.54
	ITSRN [26]	22.62M	34.19	29.46	26.22	23.96	22.64	21.80	21.26	20.87	20.56
	LTE [8]	22.53M	34.49	29.60	26.34	24.06	22.67	21.81	21.28	20.90	20.59
	BTC (ours)	22.40M	34.48	29.56	26.30	24.09	22.69	21.84	21.29	20.90	20.61
SIQAD (n = 22)	Bicubic	-	22.89	20.66	19.70	19.18	18.79	18.46	18.20	17.94	17.68
	RDN [29]	21.97M	33.53	26.89	23.38	-	-	-	-	-	-
	MetaSR [5]	22.42M	34.12	28.40	23.55	21.18	20.18	19.63	19.25	18.94	18.65
	LIIF [4]	22.32M	34.31	28.27	23.44	21.16	20.25	19.70	19.36	19.02	18.70
	ITSRN [26]	22.62M	34.68	29.07	24.03	21.44	20.38	19.77	19.40	19.09	18.79
	LTE [8]	22.53M	35.07	29.33	24.21	21.52	20.39	19.78	19.43	19.11	18.81
	BTC (ours)	22.40M	34.91	29.36	24.25	21.57	20.43	19.82	19.45	19.11	18.84

Table 1. **Quantitative comparison on SCI1K test set, SCID, and SIQAD (PSNR (dB)).** All methods are trained on SCI1K train set. The best and second results are in red and blue, respectively. RDN [29] trains different models for each scale. MetaSR [5], LIIF [4], ITSRN [26], LTE [8], and BTC use one model for all scales, and the five models utilize RDN [29] as an encoder. The number of training parameters of RDN [29] is estimated without its upsampling layer.

PSNR (dB)	non-integer scale				large scale		
Method	**×5.55**	**×6.66**	**×7.77**	**×8.88**	**×18**	**×24**	**×36**
MetaSR [5]	27.04	24.99	23.47	22.45	19.55	18.81	17.81
LIIF [4]	27.02	24.92	23.48	22.49	19.63	18.85	17.85
ITSRN [26]	27.02	24.88	23.37	22.41	19.61	18.85	17.84
LTE [8]	27.22	24.96	23.42	22.50	19.67	18.90	17.88
BTC (ours)	27.31	25.14	23.64	22.61	19.67	18.90	17.88

Table 2. **Quantitative comparison on SCI1K test set for non-integer scales and large scales (PSNR (dB)).** All methods are trained on SCI1K train set with RDN [29] encoder.

Acc./Conf. (%)	SCID (c=100)		SIQAD (c=50)	
Method	**×4**	**×5**	**×4**	**×5**
MetaSR [5]	90.89/98.19	87.07/94.89	86.55/99.08	93.55/98.53
LIIF [4]	91.00/98.26	87.17/94.69	85.23/99.23	91.56/99.13
ITSRN [26]	90.89/98.13	87.07/96.80	84.66/99.20	91.32/97.24
LTE [8]	90.89/98.00	87.07/96.18	86.36/98.98	92.31/98.90
BTC (ours)	93.63/98.55	89.70/97.88	86.93/99.32	94.29/99.22

Table 3. **Scene text recognition (STR) comparison (Prediction accuracy/Confidence (%)).** We train all methods on SCI1K train set with RDN [29] encoder and utilize a pre-trained STR network [2]. Per each scale (×4 and ×5), we randomly crop 100 and 50 text regions (32×128 sized) from SCID and SIQAD, respectively.

4. Method

4.1. Overall pipeline

Our method consists of three steps, as shown in Fig. 2.

Step 1: For a given LR image $I_{LR} \in \mathbb{R}^{3\times H\times W}$, our deep SR encoder extracts a feature of LR, $V \in \mathbb{R}^{D\times H\times W}$. The following coefficient estimator and knot estimator predict the coefficient features and knot features for scaling and translating B-spline basis functions, respectively. The coefficient estimator is a 3×3 convolutional layer with C output channels. The knot estimator is a 3×3 convolutional layer with $2\sqrt{C}$ output channels. $\sqrt{C}$ channels for X-axis knots and the rest for Y-axis knots. The dilation estimator, implemented by a single fully connected layer with $\sqrt{C}$ output dimensions, predicts dilation features from the cell size. Since the predicted coefficients and knots span an $r\times r$ local region in an HR domain, we upscale the coefficient and knot feature maps by a nearest-neighborhood spatial interpolation ($V^{\uparrow}_{coef} \in \mathbb{R}^{C\times rH\times rW}$ and $V^{\uparrow}_{knot} \in \mathbb{R}^{2\sqrt{C}\times rH\times rW}$). We utilize RDN [29] as an encoder in our experiments, removing the last upsampling layer. The channels D and C are 64 and 256, respectively.

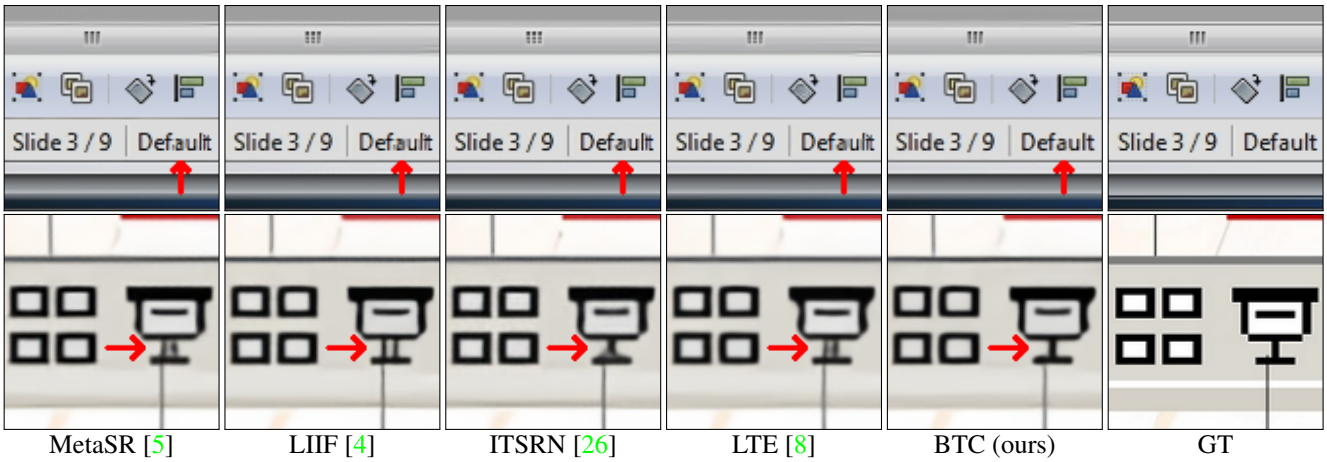


Figure 4. **Visual comparison with other arbitrary-scale SR methods** [4, 5, 8, 26] for the fraction number scales: $\times 1.49$ (top) and $\times 6.25$ (bottom). All methods are trained on SCI1K train set with RDN [29] encoder.

Method	Mem.	Time (mean±std)
MetaSR [5]	15.1 GB	655.91±2.09 ms
LIIF [4]	12.3 GB	1024.02±3.33 ms
ITSRN [26]	21.0 GB	1185.45±4.95 ms
LTE [8]	8.7 GB	1099.34±2.54 ms
BTC (ours)	7.8 GB	958.77±3.28 ms

Table 4. **Memory consumption and computation time comparison.** We use a 480×480 sized input and 960^2 query points per inference ($\times 2$ SR). The mean and std of computation time were measured with 300 iterations. All methods use RDN [29] encoder.

Step 2: Hereafter, we utilize B-spline representation with the query point's feature. To translate the B-spline basis on the coordinate around the LR pixel, we perform the element-wise subtraction between the knot features ($V^{\uparrow}_{knot}[q] \in \mathbb{R}^{2\times\sqrt{C}}$) and local coordinates ($\boldsymbol{\delta}_{xy}[q] \in \mathbb{R}^{2\times 1}$) for each XY-axis. This results are element-wise multiplied by the predicted dilations and are fed into the B-spline bases (β^n). Finally, as in Eq. (6), we conduct tensor product between X- and Y-axis bases ($\beta^n_y \otimes (\beta^n_x)^T$), and perform the element-wise multiplication between the flattened tensor product results ($\mathbb{R}^{\sqrt{C}\times\sqrt{C}} \mapsto \mathbb{R}^C$) and the predicted coefficients ($V^{\uparrow}_{coef}[q] \in \mathbb{R}^C$).

Step 3: The following INR decoder (f_θ) infers the query point's RGB values with the B-spline representation results. Our decoder is a 4-layered MLP with ReLU activations and C hidden dimensions. In detail, our method predicts the query point's RGB values by adding the decoder's output to the bilinear upscaled LR image's value as in Eq. (9).

4.2. B-spline backpropagation

The B-spline basis function $\beta^n(x)$ is a piecewise function, where n is its polynomial degree, and is set to the n^{th} convolution between $\beta^0(x)$, itself. The $\beta^0(x)$ is defined as 1 if $|x| < 0.5$ and 0 for otherwise. The $\beta^1(x)$ is supported in [-1, 1], the $\beta^2(x)$ is supported in [-1.5, 1.5], and the $\beta^3(x)$ is supported in [-2, 2]. We design BTC with $\beta^3(x)$. The $\beta^3(x)$ and differentiated $\frac{\partial}{\partial x}\beta^3(x)$ (for backpropagation) are as follows:

$$\beta^3(x) = \begin{cases} \frac{1}{6}(2+x)^3 & \text{if } -2 < x \le -1; \\ \frac{1}{6}(4-6x^2-3x^3) & \text{if } -1 < x \le 0; \\ \frac{1}{6}(4-6x^2+3x^3) & \text{if } 0 < x \le 1; \\ \frac{1}{6}(2-x)^3 & \text{if } 1 < x < 2; \\ 0 & \text{otherwise,} \end{cases} \tag{10}$$

$$\frac{\partial}{\partial x}\beta^3(x) = \begin{cases} \frac{1}{2}(2+x)^2 & \text{if } -2 < x \le -1; \\ -2x-1.5x^2 & \text{if } -1 < x \le 0; \\ -2x+1.5x^2 & \text{if } 0 < x \le 1; \\ -\frac{1}{2}(2-x)^2 & \text{if } 1 < x < 2; \\ 0 & \text{otherwise.} \end{cases} \tag{11}$$

4.3. Training strategy

Let B be the batch size, and H, W denote a training patch's height and width. First, in order to simulate arbitrary magnifications, we sample B random scales $r_{1\sim B}$ in a uniform distribution $U(1, 4)$ and crop B patches with sizes $\{r_i H \times r_i W\}_{i=1}^B$ from HR training images. Then, we randomly pick XY query points (coordinate-RGB pairs) from each HR patch in the batch from ground truth (GT) and make B LR counterparts with sizes $\{H \times W\}_{i=1}^B$ by downsampling HR patches in the scale factor $r_{1\sim B}$. We use the LR patches as the encoder's input and interpolate only query points during training. Finally, we compute the loss between the query and GT pixels. For verifying the generalization ability of our network, we evaluate our BTC for $\times 1 \sim \times 4$ (*in-training-scale*), and also $\times 5 \sim \times 10$ (*out-of-training-scale*).

PSNR (dB)	In-scale			Out-of-scale		
Method	**×2**	**×3**	**×4**	**×5**	**×7**	**×9**
BTC	39.17	**34.58**	**31.10**	**28.33**	**24.47**	**22.48**
BTC(-C)	39.06	34.56	30.96	28.23	24.31	22.35
BTC(-K)	33.90	30.09	27.42	25.83	23.10	21.57
BTC(-D)	39.05	34.54	30.94	28.19	24.42	22.41
BTC(-L)	39.01	34.04	30.49	27.95	24.34	22.36
BTC(β^2)	39.12	34.46	30.91	28.22	24.36	22.42
BTC(β^4)	**39.18**	34.57	31.04	28.25	24.41	22.45

Table 5. **Ablation study on SCI1K test set (PSNR (dB)).** Definitions of -C/K/D/L and $\beta^{2/4}$ are described in Sec. 5.3. All methods are trained on SCI1K train set with RDN [29] encoder. The best results are bolded.

5. Experiments

5.1. Training

Dataset To demonstrate the ability of our model, we utilize SCI1K dataset [26] for SCI SR experiments. It consists of 1000 screenshots with 800 images for the train set and 200 for the testing set and we follow the standard split [26] for train and test set. We also report the results on two other screen content datasets: SCID [13] and SIQAD [25]. We train all the methods only on SCI1K train set.

Implementation details For training, we use LR patches with the sizes of $\{48 \times 48\}_{i=1}^{16}$ downsampled by bicubic resizing, and augment the patches with flipping and rotations. Following [29], we utilize L1 loss and the Adam [7] optimizer with $\beta 1 = 0.9$ and $\beta 2 = 0.999$. The networks are trained for 1000 epochs with batch size 16, and the learning rate initialized as 1e-4 is decayed in half every 200 epochs.

5.2. Evaluation

Quantitative results Tab. 1 demonstrates a quantitative comparison between our method, RDN [29], and existing arbitrary-scale SR methods (MetaSR [5], LIIF [4], ITSRN [26], LTE [8]) on SCI1K test set, SCID, and SIQAD. All the methods are trained on the train set of SCI1K. Since RDN [29] is dedicated to an upscaling factor, each model on the specific scale needs to be trained. Except for SCID in-training-scales and SIQAD×2, our method shows the best performance with competitive model size for almost every scale factor and dataset. The maximum gain is up to 0.17dB for ×4 on SCI1K. Moreover, BTC outperforms ITSRN [26] by 0.43dB, 0.26dB, and 0.28dB at SCI1K×2, ×3, and ×4 with 220K fewer parameters. Compared to LTE [8], BTC performs better by 0.03dB, 0.08dB, and 0.17dB at SCI1K×2, ×3, and ×4 with 130K fewer parameters. In Tab. 2, our method gains 0.09dB, 0.18dB, 0.16dB, and 0.11dB at ×5.55, ×6.66, ×7.77, and ×8.88, correspondingly. For large scales, BTC shows better performance over MetaSR [5], LIIF [4], and ITSRN [26].

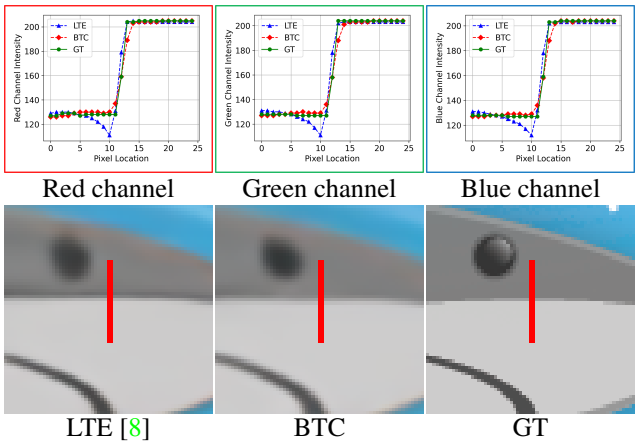


Figure 5. **Robustness of BTC against undershooting compared to LTE** [8] (×8). The graphs in the first row present the red, green, and blue channel intensity along the red lines in the second row.

Qualitative results Figs. 3 and 4 present a qualitative comparison with other arbitrary-scale SR methods [4, 5, 8, 26] for integer number scales (×3, ×4, ×5, and ×7) and fraction number scales (×1.49 and ×6, 25), respectively. As shown in Fig. 3, our method represents the thin edges of characters and reconstructs the information of texts better than other methods. Moreover, the pre-trained scene text recognition algorithm [2] recognizes our reconstructed characters more accurately and confidently. In Fig. 4, BTC restores texts and graphics better than other methods, even on the fractional number scales.

5.3. Ablation study

For ablation study, we retrain several configurations with RDN [29] encoder: BTC(-C), BTC(-K), BTC(-D), and BTC(-L) indicate BTC without a coefficient estimator, a knot estimator, a dilation estimator, and an LR skip connection, respectively. The B-spline basis's polynomial degree of BTC($\beta^{2/4}$) is 2/4. Tab. 5 shows that a coefficient estimator, a dilation estimator, and an LR skip connection contribute 0.1dB, 0.12dB, and 0.57dB at ×4 SR, respectively. Note that the gain of a knot estimator is significantly higher than the gain of the other things. Moreover, we can observe that β^3 is superior to $\beta^{2/4}$.

5.4. Computation cost comparison

Tab. 4 compares our method's memory consumption and computation time with other arbitrary-scale SR methods on NVIDIA RTX 3090 24GB. We compute the memory consumption and computation time with the ×2 SR task (input: 480×480). Our BTC has competitive memory consumption compared to MetaSR and ITSRN. MetaSR estimates the filter's weights for each pixel, so it performs the matrix product between the filters and the LR feature, consuming more memories.

Train set: DIV2K		In-scale			Out-of-scale	
Test set	Method	×2	×3	×4	×6	×8
Set5 (n = 5)	BTC	**38.26**	**34.73**	32.57	29.26	27.23
	LTE [8]	38.23	34.72	**32.61**	**29.32**	**27.26**
Set14 (n = 14)	BTC	34.05	30.58	28.84	26.67	**25.21**
	LTE [8]	**34.09**	30.58	**28.88**	**26.71**	25.16
B100 (n = 100)	BTC	32.36	29.30	**27.78**	26.01	24.95
	LTE [8]	32.36	29.30	27.77	26.01	24.95
Urban100 (n = 100)	BTC	32.98	28.95	26.81	24.28	22.88
	LTE [8]	**33.04**	**28.97**	26.81	24.28	22.88

Table 6. **Quantitative limitation of BTC compared to LTE [8] on natural image benchmarks: Set5, Set14, B100, and Urban100 (PSNR (dB))**. All methods are trained on DIV2K train set with RDN [29] encoder. The best results are bolded.

Similarly, ITSRN utilizes an MLP to model an implicit transformer, including a matrix product between features, causing more memory consumption. On the other hand, BTC uses convolutional layers to learn B-spline features and reduces the number of convolution filter channels by the tensor product. Meanwhile, MetaSR does not perform local ensemble, so the computation time is faster than LIIF, LTE, and BTC. However, blocking artifacts may occur.

6. Discussion

Advantage of B-spline over Fourier information LTE [8] represents the signal as Fourier series close to the HR image's Fourier features. The finite sum of sinusoids, caused by finite size of channels, leads to convolution with *sinc* function (ideal low-pass filter on *freq.* domain). Then, we are able to reformulate the signal representation in LTE as follow:

$$\underbrace{s(x) = \sum_{t \in \mathcal{I}} c[t] sinc\left(x - k'[t]\right)}_{\text{LTE's Fourier representation}} \Leftrightarrow \underbrace{s(x) = \sum_{t \in \mathcal{I}} c[t] \beta^n \left(\frac{x - k[t]}{d[t]}\right)}_{\text{BTC's B-spline representation}},$$

where $sinc(x) = \frac{\sin(x)}{x}$, k' is the uniformly distributed translations, and $\mathcal{I}$ is a set of index. Here, the rippled side-lobes of $sinc(\cdot)$ can cause under/overshooting at the discontinuities (similar to Gibb's Phenomena). Because screen content images include many texts and graphics, the edges of such contents cause lots of discontinuities.

In BTC (right-hand side), the positive B-spline kernels (integrated to 1) make the values between the minimum and maximum of the input signal. The smoothing effect of the B-spline kernel is reduced by dynamically allocating the weights and translations to $\beta^n(\cdot)$ with compact support. As shown in Fig. 5, BTC is robust to undershooting compared to LTE at the sharp edges. LTE causes undershooting aliasing at a discontinuity (the black pixels along the border on the shark's face in LTE's SR result).

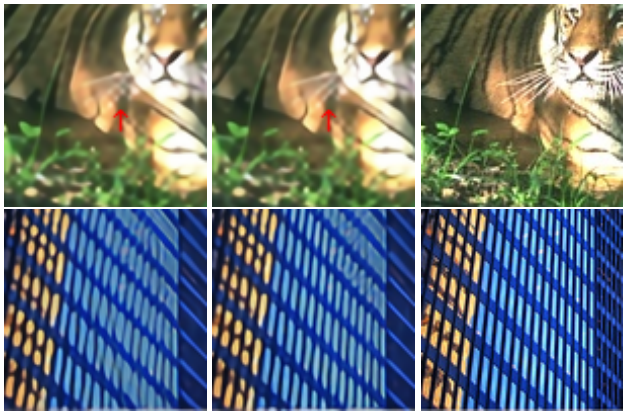


Figure 6. **Qualitative limitation of BTC compared to LTE [8] on natural image benchmarks (×3): B100 (top) and Urban100 (bottom).** All methods are trained on DIV2K train set with RDN [29] encoder.

Limitation on natural images Although BTC performs better on screen content images, BTC is not always superior to LTE. We compare our method with LTE [8], training both methods on DIV2K [1] train set. From Tab. 6, BTC shows the same or lower performance as LTE on natural image benchmarks (Set5 [3], Set14 [27], B100 [11], and Urban100 [6]). Since natural images mainly contain a lot of continuous and repetitive textures, LTE, which represents signals with Fourier basis, generally performs better on natural images as shown in Fig. 6.

7. Conclusion

This paper proposes a B-spline Texture Coefficients estimator (BTC) for arbitrary scale SCI SR. Our BTC-based SR method achieves the best performance with a competitive model size for screen content datasets. Furthermore, our method restores the thin edges of text or graphics better than other arbitrary-scale SR methods. Compared with the LTE utilizing Fourier representation, BTC has fewer ringing artifacts caused by overshooting or undershooting owing to the compactly and positively supported B-spline. Moreover, our BTC shows efficient memory consumption and computation time. Our SR results are recognized as correct text letters with the highest confidence by a pre-trained scene text recognition network.

Acknowledgement This work was partly supported by the National Research Foundation of Korea (NRF) grant funded by the Korea government (No. 2021R1A4A1028652), the DGIST R&D Program of the Ministry of Science and ICT (21-IJRP-01), Smart HealthCare Program(www.kipot.or.kr) funded by the Korean National Police Agency(KNPA) (No. 230222M01), and Institute of Information & communications Technology Planning & Evaluation (IITP) grant funded by the Korea government (MSIT) (No.2021-0-02068, Artificial Intelligence Innovation Hub).

References

[1] Eirikur Agustsson and Radu Timofte. NTIRE 2017 Challenge on Single Image Super-Resolution: Dataset and Study. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*, July 2017. 8

[2] Darwin Bautista and Rowel Atienza. Scene text recognition with permuted autoregressive sequence models. In *European Conference on Computer Vision*, pages 178–196. Springer, 2022. 4, 5, 7

[3] Marco Bevilacqua, Aline Roumy, Christine Guillemot, and Marie line Alberi Morel. Low-Complexity Single-Image Super-Resolution based on Nonnegative Neighbor Embedding. In *Proceedings of the British Machine Vision Conference*, pages 135.1–135.10. BMVA Press, 2012. 8

[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, June 2021. 1, 2, 3, 4, 5, 6, 7

[5] Xuecai Hu, Haoyuan Mu, Xiangyu Zhang, Zilei Wang, Tieniu Tan, and Jian Sun. Meta-SR: A Magnification-Arbitrary Network for Super-Resolution. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2019. 1, 4, 5, 6, 7

[6] Jia-Bin Huang, Abhishek Singh, and Narendra Ahuja. Single Image Super-Resolution From Transformed Self-Exemplars. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2015. 8

[7] Diederik P. Kingma and Jimmy Ba. Adam: A Method for Stochastic Optimization. In Yoshua Bengio and Yann LeCun, editors, *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*, 2015. 7

[8] Jaewon Lee and Kyong Hwan Jin. Local Texture Estimator for Implicit Representation Function. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1929–1938, 2022. 1, 2, 3, 4, 5, 6, 7, 8

[9] Jingyun Liang, Jiezhang Cao, Guolei Sun, Kai Zhang, Luc Van Gool, and Radu Timofte. SwinIR: Image Restoration Using Swin Transformer. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV) Workshops*, pages 1833–1844, October 2021. 1

[10] 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 Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*, July 2017. 1

[11] 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*, volume 2, pages 416–423 vol.2, 2001. 8

[12] Ben Mildenhall, Pratul P. Srinivasan, Matthew Tancik, Jonathan T. Barron, Ravi Ramamoorthi, and Ren Ng. NeRF: Representing Scenes as Neural Radiance Fields for View Synthesis. In *Proceedings of the European Conference on Computer Vision (ECCV)*, August 2020. 2, 3

[13] Zhangkai Ni, Lin Ma, Huanqiang Zeng, Jing Chen, Canhui Cai, and Kai-Kuang Ma. Esim: Edge similarity for screen content image quality assessment. *IEEE Transactions on Image Processing*, 26(10):4818–4831, 2017. 7

[14] Anjana Deva Prasad, Aditya Balu, Harshil Shah, Soumik Sarkar, Chinmay Hegde, and Adarsh Krishnamurthy. NURBS-Diff: A Differentiable Programming Module For NURBS. *Computer-Aided Design*, page 103199, 2022. 2

[15] Nasim Rahaman, Aristide Baratin, Devansh Arpit, Felix Draxler, Min Lin, Fred Hamprecht, Yoshua Bengio, and Aaron Courville. On the spectral bias of neural networks. In *International Conference on Machine Learning*, pages 5301–5310. PMLR, 2019. 1, 2

[16] Isaac Jacob Schoenberg. Contributions to the problem of approximation of equidistant data by analytic functions. part b. on the problem of osculatory interpolation. a second class of analytic approximation formulae. *Quarterly of Applied Mathematics*, 4(2):112–141, 1946. 2

[17] Vincent Sitzmann, Julien Martel, Alexander Bergman, David Lindell, and Gordon Wetzstein. Implicit Neural Representations with Periodic Activation Functions. In H. Larochelle, M. Ranzato, R. Hadsell, M. F. Balcan, and H. Lin, editors, *Advances in Neural Information Processing Systems*, volume 33, pages 7462–7473. Curran Associates, Inc., 2020. 2

[18] Matthew Tancik, Pratul Srinivasan, Ben Mildenhall, Sara Fridovich-Keil, Nithin Raghavan, Utkarsh Singhal, Ravi Ramamoorthi, Jonathan Barron, and Ren Ng. Fourier Features Let Networks Learn High Frequency Functions in Low Dimensional Domains. In H. Larochelle, M. Ranzato, R. Hadsell, M. F. Balcan, and H. Lin, editors, *Advances in Neural Information Processing Systems*, volume 33, pages 7537–7547. Curran Associates, Inc., 2020. 2, 3

[19] Michael Unser. Splines: A perfect fit for signal and image processing. *IEEE Signal processing magazine*, 16(6):22–38, 1999. 2

[20] Michael Unser, Akram Aldroubi, and Murray Eden. Fast B-spline transforms for continuous image representation and interpolation. *IEEE Transactions on pattern analysis and machine intelligence*, 13(3):277–285, 1991. 2

[21] Michael Unser, Akram Aldroubi, and Murray Eden. B-spline signal processing. I. Theory. *IEEE transactions on signal processing*, 41(2):821–833, 1993. 2, 3, 4

[22] Michael Unser, Akram Aldroubi, and Murray Eden. B-spline signal processing. II. Efficiency design and applications. *IEEE transactions on signal processing*, 41(2):834–848, 1993. 2

[23] Peng-Shuai Wang, Yang Liu, Yu-Qi Yang, and Xin Tong. Spline positional encoding for learning 3d implicit signed distance fields. *arXiv preprint arXiv:2106.01553*, 2021. 2, 3

[24] Keyulu Xu, Mozhi Zhang, Jingling Li, Simon Shaolei Du, Ken-ichi Kawarabayashi, and Stefanie Jegelka. How Neural Networks Extrapolate: From Feedforward to Graph Neural Networks. In *9th International Conference on Learning Representations, ICLR 2021, Virtual Event, Austria, May 3-7, 2021*. OpenReview.net, 2021. 4

[25] Huan Yang, Yuming Fang, and Weisi Lin. Perceptual quality assessment of screen content images. *IEEE Transactions on Image Processing*, 24(11):4408–4421, 2015. 1, 2, 7

[26] Jingyu Yang, Sheng Shen, Huanjing Yue, and Kun Li. Implicit Transformer Network for Screen Content Image Continuous Super-Resolution. *Advances in Neural Information Processing Systems*, 34:13304–13315, 2021. 1, 2, 4, 5, 6, 7

[27] Roman Zeyde, Michael Elad, and Matan Protter. On Single Image Scale-Up Using Sparse-Representations. In Jean-Daniel Boissonnat, Patrick Chenin, Albert Cohen, Christian Gout, Tom Lyche, Marie-Laurence Mazure, and Larry Schumaker, editors, *Curves and Surfaces*, pages 711–730, Berlin, Heidelberg, 2012. Springer Berlin Heidelberg. 8

[28] Yulun Zhang, Kunpeng Li, Kai Li, Lichen Wang, Bineng Zhong, and Yun Fu. Image Super-Resolution Using Very Deep Residual Channel Attention Networks. In *Proceedings of the European Conference on Computer Vision (ECCV)*, September 2018. 1

[29] Yulun Zhang, Yapeng Tian, Yu Kong, Bineng Zhong, and Yun Fu. Residual Dense Network for Image Super-Resolution. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2018. 1, 4, 5, 6, 7, 8