Single-Image Super-Resolution Analysis in DCT Spectral Domain

O. AYDIN and R.G. CINBIS

***Abstract*—Advances in deep learning techniques have lead to drastic changes in contemporary methods used for a diverse number of computer vision problems. Single-image super-resolution is one of these problems that has been significantly and positively influenced by these trends. The mainstream state-of-the-art methods for super-resolution learn a non-linear mapping from low-resolution images to high-resolution images in the spatial domain, parameterized through convolution and transposed-convolution layers. In this paper, we explore the use of spectral representations for deep learning based super-resolution. More specifically, we propose an approach that operates in the space of discrete cosine transform based spectral representations. Additionally, to reduce the artifacts resulting from spectral processing, we propose to use a noise reduction network as a post-processing step. Notably, our approach allows using a universal super-resolution model for a range of scaling factors. We evaluate our approach in detail through quantitative and qualitative results.**

***Index Terms*—Super resolution, deep learning, image process.**

I. Introduction

THE primary goal of single-image super-resolution (SR) is to reconstruct a high-resolution (HR) image from a single low-resolution (LR) image with maximum perceptual affinity. Single-image SR has recently attracted a great interest due to its possible applications in a variety of areas, including medical imaging, remote sensing, consumer photo enhancement, and video surveillance. However, SR remains as an unsolved problem mainly due to its ill-posed nature: there can be infinitely many scenes yielding the same LR image. Therefore, in SR, the goal is to find the perceptually most plausible HR image(s) corresponding to a given LR image.

From a machine learning perspective, LR-to-HR mapping is a regression problem. To tackle this problem, a variety of traditional machine learning approaches have previously been proposed, such as local linear regression [1], dictionary learning [2] and random forests [3]. More recently, the progress in SR has been dominated by deep learning (DL) based approaches, leading to significant improvements in the state-of-the-art models thanks to learning better nonlinear mappings, e.g. [4]–[6].

ONUR AYDIN, is with the Department of Computer Engineering, Bilkent University, Ankara, 06800, Turkey, (e-mail: onuralg@gmail.com).
https://orcid.org/0000-0002-9304-0647

RAMAZAN GOKBERK CINBIS, is with the Department of Computer Engineering, METU, Ankara, 06800, Turkey, (e-mail: gcinbis@ceng.metu.edu.tr).
https://orcid.org/0000-0003-0962-7101

Manuscript received April 03, 2020; accepted July 14, 2020.
DOI: 10.17694/bajece.714293

Two mainstream ways of utilizing deep learning techniques in super-resolution problem are available. In the first approach, the input image is resized to the target scale using a basic method, such as bicubic interpolation. Then, the SR problem degrades to learning a non-linear transformation that enhances the image quality of the HR image. In the second one, the upscaling transform is directly learned within the deep learning architecture, typically using *transposed convolution* layer(s) [7]. In both cases, LR - HR image pairs are typically required during model training. Once the training is complete, the model is used to predict the HR versions of novel LR image inputs.

In designing an SR approach, there are arguably three primary concerns. *Precision* which demonstrates how accurately target high-resolution image is reconstructed is the main factor. The second one is *efficiency*, which refers to the inference-time computational requirements. Efficiency can especially be critical in applications requiring real-time processing and/or inference on low-power devices. The third one is *flexibility* in terms of selecting an output scale factor at test time, which determines the area ratio between the output and input image. Utilizing a separate network for every scale factor is inherently costly, inefficient and impractical. Furthermore, since model training gets more difficult as the scale factor increases, it is also not plausible to learn a model that is trained only for the largest scaling factor of interest and then down-scale from its output as needed.

To tackle the SR problem, we focus on the use of frequency domain deep learning approaches. The frequency-based representations are relatively little studied in the domain of deep learning. A prominent study in this area is Rippel et al. [8], which shows that convolutional neural networks (CNNs) can be used to learn image classification models in the Fourier domain. Wang et al. [9] shows that discrete cosine transform (DCT) can be used to compress weights of CNNs while preserving the prediction accuracy. Kumar et al. [10] shows that CNNs can be trained to predict wavelet coefficients to improve SR performance. Only in recent works [11] and [12] Fourier domain CNNs have been explored for SR.

In our work, we are approaching single-image super-resolution problem via learning a deep neural network in DCT based frequency domain. More specifically, we train deep neural networks to learn how to transform input low-resolution images into high resolution ones, within the DCT frequency representation. Then, in the spatial domain, a pre-trained artifact reduction model is utilized to eliminate unintended effects appearing when the resulting frequency domain representation is transformed back to the spatial domain. We comprehensively

evaluate our approach on benchmark datasets and discuss its benefits and drawbacks. Our formulation provides us a fast and efficient approach, and, gets rid of the necessity to train a separate model for arbitrary scale factors, i.e. a single model can be used for super resolution to a variety of scaling factors.

Outline. In Section II, we provide a brief overview of deep learning based single-image SR. In Section III, we present our analysis on the use of DCT domain for SR and the details of our approach. In Section IV, we provide our experimental results with detailed evaluations and comparisons to contemporary deep SR approaches. In Section V, we conclude with a summary of our observations and discussion on possible future work directions.

II. Related work

In this section, we present an overview of well-known spatial domain and recent frequency domain approaches for SR. A more comprehensive overview of deep learning based methods for SR can be found in the recent survey by Anwar et al. [7].

Super-Resolution Convolutional Neural Network (SRCNN) [4] proposes one of the first deep learning architectures for single-image SR. It applies a 3-layer CNN on the output of bicubic interpolation output. The model is trained using ℓ_2 reconstruction loss between the SR output and the target HR image. One disadvantage of the SRCNN model is that for every scaling factor, a different model is trained. While SRCNN is not the state-of-the-art on benchmark datasets anymore, it is still a good reference for DL-based SR due to its simplicity.

Several papers propose improvements over the SRCNN approach. For example, Faster Super-Resolution Convolutional Neural Network (FSRCNN) [13] proposes a deeper architecture that uses a transposed convolution layer, instead of upsampling using bicubic interpolation as a preprocessing step. More specifically, in FSRCNN architecture, seven convolutional layers and single transposed-convolution layer is used. Like SRCNN, the model is trained over the ℓ_2 reconstruction loss. Very Deep Super Resolution (VDSR) [5] improves the SRCNN architecture by stacking 20 convolutional layers and adding residual connections. In addition, using *scale augmentation*, the approach trains one model for all scaling factors. Super-Resolution Generative Adversarial Networks (SRGAN) [14] uses adversarial training for improving SR outputs. The SR model contains a *generator*, and a *discriminator* network is used to enforce the generator to produce SR outputs indistinguishable from real HR images. The approach uses *perceptual loss*, i.e. reconstruction loss in convolutional feature space, in addition to the adversarial loss. Laplacian Super-Resolution Networks (LapSRN) [6] proposes to progressively increase the image resolution over a Laplacian pyramid, via 27 convolutional layers with residual connections. The model is trained using *Charbonnier loss*, which is a robust reconstruction loss function that handles outliers better than ℓ_2 loss.

In the recent work of Dai et al. [15], it is highlighted that most SR approaches neglect correlations of intermediate layers. This work proposes the *second-order attention network* to model correlations in intermediate layers and to learn more discriminative representations by adaptive re-scaling of features. Additionally, a non-locally enhanced residual group scheme is proposed in order to capture long-distance spatial information and local-source residual attention groups are proposed to learn abstract feature representations.

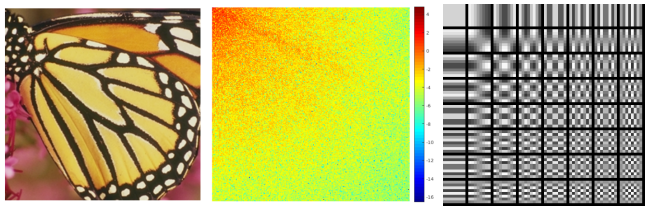
Another open problem in SR is effective training on deep architectures, especially for large scaling factors. Towards tackling this problem, Wang et al. [16] proposes a progressive learning approach. Combined with an adversarial training scheme, this method obtains significant improvements, especially for high scaling factors.

Kumar et al. [10] proposes the Wavelet Domain Super-Resolution (CNNWSR) model, which aims to directly predict discrete wavelet transform coefficients of the high-resolution target image. The predicted wavelet coefficients are utilized to reconstruct high-resolution images via two-dimensional inverse DWT. Unlike SRCNN which reconstructs a single-image, CNNWSR architecture is using convolutional layers to predict three separate images that contain all wavelet coefficients. This architecture is the first solution that fuses the deep learning and spectral approaches for SR problem. Nevertheless, the core issue of the CNNWSR architecture is limited to select arbitrary scale factors which is caused from the nature of wavelet transform. Frequency Domain Super Resolution (FNNSR) [11] and Improved-FNNSR (IFNNSR) [12] are two recently proposed Fourier domain SR approaches. FNNSR formulates a deep neural network that parameterizes convolutions as point-wise multiplications in the spectral domain using single convolutional layers to approximate ReLU activations. IFNNSR improves this approach mainly by (i) using Hartley transform instead of Fourier transform, (ii) utilizing multiple convolutional layers to better approximate ReLU activations and (iii) proposing a novel weighted Euclidean loss that emphasizes the errors at high frequency components.

There are a couple of major differences between our approach and the aforementioned frequency domain SR approaches. First of all, while we train a fully connected super-resolution network for predicting carefully-selected frequency terms, other frequency domain approaches define convolutional networks in the frequency domain and predict all frequency terms. Secondly, while we are training a single model for any scale factor, in other spectral-domain SR methods, a separate network is needed for each scale factor. Third, we use DCT as the spectral representation, mainly for its well-known representational power that does not require complex numbers.

III. Method

In this section, we first give a summary of the discrete cosine transform (DCT). In Section III-B, we present our analysis and observations on DCT based super-resolution. Then, in Section III-C, we present our DCT-based super-resolution approach in detail. Finally, in Section III-D, we present the post-processing approach based on an artifact reduction network that aims to eliminate the ringing artifacts.



(a) (b)

Figure 1: (a) 2D DCT example, (b) 2D DCT bases shown as images.

A. Discrete Cosine Transform

The spectral domain transforms convert a time (spatial) domain signal to the frequency domain without any information loss. Arguably, the most well-known discrete transform is the Discrete Fourier Transform (DFT). However, even if the input signal is real, DFT yields a representation involving complex numbers. While it is possible to handle complex numbers in a compute graph, the involvement of complex number arithmetic naturally introduces complexity, which may lead to difficulties especially when deploying to low-cost inference devices.

Therefore, in our work, we focus on the Discrete Cosine Transform (DCT). DCT decomposes a signal into cosine functions oscillating at various frequencies and yields only real-valued numbers in the spectral representation. For a two dimensional discrete signal f and its frequency domain representation F, two-dimensional DCT is described as follows [17]:

$$F[u,v] = a(u)a(v)\sum_{x=0}^{N-1}\sum_{y=0}^{M-1} f[x,y]\gamma(x,y,u,v) \quad (1)$$

where $\gamma(x,y,u,v)$ is defined as

$$\gamma(x,y,u,v) = \cos\left(\frac{\pi(2x+1)u}{2N}\right)\cos\left(\frac{\pi(2y+1)v}{2M}\right) \quad (2)$$

and $a(u)$ is defined as:

$$a(u) = \left\{ \begin{array}{ll} \sqrt{\frac{1}{N}}, & u = 0 \\ \sqrt{\frac{2}{N}}, & u \neq 0 \end{array} \right\} \quad (3)$$

The signal transformed to frequency domain is reconstructed back via two-dimensional Inverse DCT which is described as follows:

$$f[x,y] = \sum_{u=0}^{N-1}\sum_{v=0}^{M-1} a(u)a(v)F[u,v]\gamma(x,y,u,v). \quad (4)$$

In the Figure 1a, 2-D DCT of an example image is provided. On the one hand, as can be seen in the figure, the resulting DCT spectral representation is difficult to understand directly due to the "loss" (i.e. transformation) of spatial structure. On the other hand, DCT spectral representation allows easily exploring the distribution of information across various frequency components. The top-left corner values correspond to lower frequencies and bottom-right corner values correspond to higher frequencies. As it can be seen through this example, a notable feature of DCT is its energy compaction property. Thanks to this property, DCT preserves substantially more information in lower frequency components [18]. In this context, the relationship between DCT and Karhunen-Loéve transform, which is known to provide the optimal bases for linear approximations of stochastic processes under certain assumptions, is notable [19].

Furthermore, in the Figure 1b, the basis functions for 2-D DCT is given. Each basis function is a 2-D representation of the mixture of two cosine functions which are oscillated at different frequencies. The first basis function located at the top left corner is the DC term. From top to bottom and from left to right, the frequencies of cosine functions increase.

B. Super-resolution in DCT spectral domain

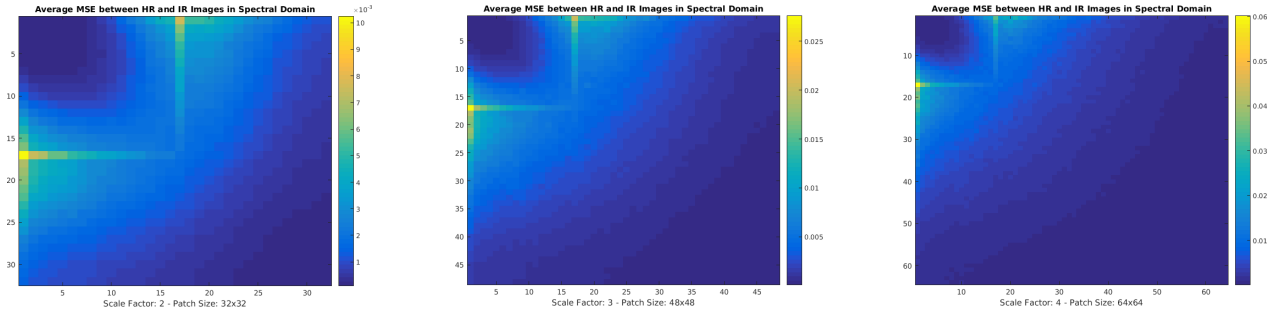
In this section, we present our analysis of the problem of SR on the DCT spectral representation domain. We use the analysis and our main observations presented in this section to design and construct our SR network.

In our approach, we utilize bicubic interpolation to resize a given low-resolution image to the target image size as a pre-processing step, e.g. we up-scale by a factor of $2\times, 3\times, 4\times$ or similar. Thereafter, the image is split to fixed-sized patches with a stride of patch size. Throughout our study, we fix the input patch size 16×16 pixels. The goal is to synthesize HR patches from these LR patches, therefore, we use bicubic interpolation to obtain initial output patches, e.g. of size $32 \times 32, 48 \times 48, 64 \times 64$ for the scaling factors $2\times, 3\times, 4\times$, respectively.

After obtaining the initial output patches, we compute their spectral representations using 2-d discrete cosine transform. Then, to better understand the problem, we compute the squared error between bicubic interpolated patches and true HR patches over the DCT coefficients of the patches, and, average these errors across a large sample of patches from our training set (see Section IV). We show the resulting mean square errors values for three different scale factors in Figure 2a, 2b and 2c. In these images, each pixel represents a frequency value. Following the ordering in Figure 1, while the top left corner stands for low frequencies, the bottom right corner stands for high frequencies and the remaining regions stand for mid-frequencies.

If we interpret these mean squared error values as distribution of error in the space of DCT coefficients, we observe that lower-mid range contains the largest problematic region (we simply refer to this region as mid-frequencies for brevity), instead of low-frequencies or higher frequencies. We also observe that this error distribution in the DCT coefficient space is consistent across all three scaling factors.

Following these observations, we focus on minimizing the coefficient errors on the most problematic DCT components by applying a neural network architecture to the DCT representations of bicubic interpolation output. We emphasize the importance of the observation that the error accumulates in the same DCT components: this leads us to construct a single model for all scaling factors, thanks to the fact that higher frequency DCT components have relatively low importance. Therefore, by focusing on only the mid-frequency terms, one



(a) 2× scaling (b) 3× scaling (c) 4× scaling

Figure 2: Mean squared error of DCT coefficients between bicubic interpolated patches and HR patches at three different scaling factors.

can deploy a single model that does can be utilized for a range of scaling factors.

C. Our super-resolution network

The primary purpose of our neural network architecture is learning a mapping from bicubic interpolated image patches to ground truth HR image patches. Additionally, we aim to make the model applicable for various scale factors. Hence, input and output dimensions shall be coherent for all scale factors. Towards meeting these goals, we observe that if the network is trained to map all DCT coefficients, as the number of coefficients varies depending on the number of inputs, the network will need to be trained for each scale factor separately. Therefore, we instead focus on specific frequency components that are available above all scaling factors larger than a minimum (2× in our case) factor.

To realize this approach, following our observations made in the previous section, we focus on the problem of mapping mid-frequency DCT coefficients of LR inputs to values closer to those of true HR patches. We select the target frequency components by finding the most problematic 512 frequency values, according to the mean square error analysis results. These frequency components naturally fall into the mid-frequencies band.

We illustrate our overall approach in Figure 3. The model first converts the input LR image into its DCT spectral representation. Then we take the selected frequency components and feed them to the neural network. Here, the binary mask shown in the figure corresponds to the 512 mid-frequency components that we truly use in our experiments. We replace these coefficients with those produced by the network and reconstruct the SR image through inverse DCT (IDCT).

In training our SR network, we use mean square error as our loss function. The error is measured between the coefficients produced by the SR network and those of the target HR patches. As our network architecture, we use a feed-forward fully connected neural network. Our choice of fully-connected layers instead of convolutional layers is motivated by the observation that local structure is repetitive and it is hard to apply the same mapping onto different regions in the spectral domain, unlike the spatial domain. In addition, the use of fully-connected layers allows us to choose arbitrary frequency components for processing, without requiring an image-like structure.

In the network architecture, we use a four-layered fully connected neural network with 512 neurons per layer. In total, the network contains 1050624 parameters. To prevent from overfitting, after every fully connected layer, we place a dropout layer [20]. Since discrete cosine transform typically yields numbers in the range from -1 to 1 (except the DC term), we use a hyperbolic tangent function as the activation function. We use Xavier initializer [21] for model initialization and use Adam optimizer [22] for optimization. We use a batch-size of 128 in training. Overall, we observe relatively little variance in training loss and validation performance scores with respect to changes made in architectural details and other hyper-parameters.

D. Artifact reduction in spatial domain

It is well known that manipulations on spectral image representations easily lead to apparent artifacts in the spatial domain. This is essentially due to the fact that manipulation of a single spectral coefficient corresponds to jointly manipulating all pixels, at varying wave-like degrees parameterized by a periodic function. Given that our SR algorithm corresponds to manipulation of only a subset of coefficients by design, we observe ringing artifacts on the SR output.

To reduce the resulting ringing artifacts, we apply an artifact reduction solution as a post-processing step. In our experiments, we use a pre-trained AR-CNN model [23], which uses an SRCNN-like convolutional neural network architecture. This network is trained to learn a mapping from JPEG compressed images to pre-compression images to learn to reduce compression artifacts. The model is trained using mean-square error, on a newly built dataset called *dataA*. We observe that even if the AR-CNN model is trained on a entirely different dataset, it leads to a significant artifact reduction on our SR image results. We show its effect through an example on Figure 4, where the intermediate steps (bicubic interpolation, spectral super-resolution and artifact reduction) of our complete single-image super-resolution approach can be seen. The corresponding improvements in PSNR scores can also be seen in this figure.

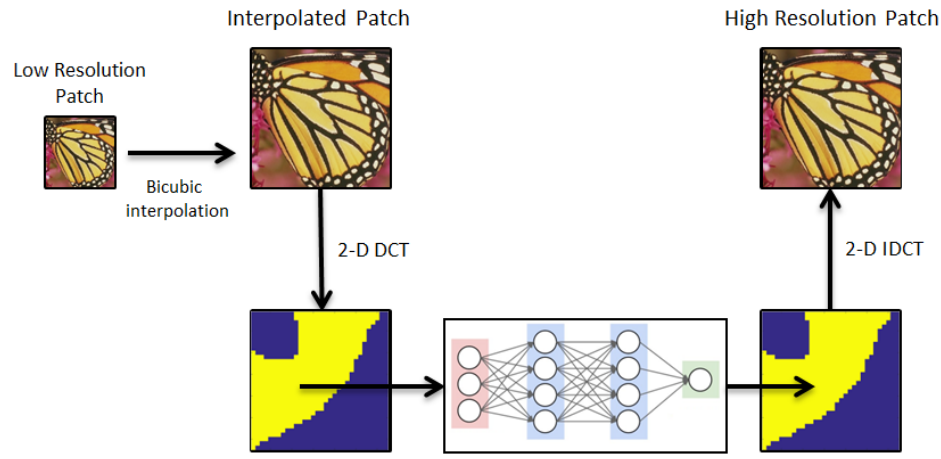


Figure 3: Our Spectral Super Resolution approach.

Algorithm 1 Spectral Super-Resolution

Input: Low-resolution input image (I_{LR}).
Input: Target HR scale (α).
Output: Predicted output image.
$I_{HR}^{Bicubic} \leftarrow$ BicubicInterpolation(I_{LR},α)
patches $\leftarrow$ SplitToPatches($I_{HR}^{Bicubic}$)
for index **in** 1 . . . Count(patches) **do**
    $S_{patch} \leftarrow$ DCT(patches[index])
    S_{patch}[FreqMask] $\leftarrow$ SRNetwork(S_{patch}[FreqMask])
    patches[index] $\leftarrow$ InverseDCT(S_{patch})
$I_{HR}^{SR} \leftarrow$ MergePatches(patches)
$I_{HR}^{PostProcessed} \leftarrow$ ARCNN(I_{HR}^{SR})
return $I_{HR}^{PostProcessed}$

We give the summary of complete super-resolution inference steps in Algorithm 1. The algorithm takes a low-resolution image and a scale factor which indicates how much the image should be resized. Initially, the input image is resized with bicubic interpolation to the given scale factor. Then, the resized image is divided into patches and each patch is processed separately. Here, each patch is first transformed into the frequency domain using discrete cosine transform, and problematic frequency regions are improved using our super-resolution network. Then, the improved patches are transformed back to the spatial domain and each patch is located back to their original place. In the post-processing step, the complete super-resolved image is given to the ARCNN model to reduce the artifacts resulting from spectral domain processing.

IV. Experiments

Our experimental setup and results with detailed analyses are given in this stage. In Section IV-A, experimental setup and implementation details are explained. In Section IV-B, we present our experimental results.

A. Experimental setup

In this section, we present (i) the details of train, validation and test datasets, (ii) the evaluation metrics and (iii) the model selection details.

Datasets. In the training phase, we use the widely used BSDS200, General100, and T91 datasets, which contain 200, 100 and 91 images, respectively. Following the common practice, e.g. [6], we use the combination of these three datasets and obtain 391 training images in total. For evaluation, we use four separate datasets: Set5 [24], Set14 [25], BSDS100 [26], and Urban100 [27]. These datasets contain 5, 14, 100 and 100 images, respectively. We use Set5 as our validation set for architecture and hyper-parameter selection and use the remaining three datasets for test evaluation.

Performance metrics. For quantitative evaluation of the SR results, we use the *Peak Signal-to-Noise Ratio* (PSNR) and *Structural Similarity Index* (SSIM) [28] evaluation metrics, following the common practice [7]. We note that while higher PSNR and SSIM values are desirable in theory, these metrics are not fully correlated with true perceptual quality [14]. Utilizing a better evaluation metric for the super-resolution problem remains as an open problem in SR.

Model selection. We use performance scores on the validation set for architecture and hyper-parameter tuning. To optimize over the hyper-parameter combinations, we use grid search over the learning rate and the number of neurons per layer. As the learning rate candidates, we use the set $\{10^{-4}, 10^{-5}, 10^{-6}\}$. As the number of neurons per layer (i.e.number of hidden units), we use the candidate set $\{256, 512, 1024\}$.

In our experiments, we have obtained very similar training curves across varying learning rates and varying number of neurons per layer. We have observed that while training speeds up for higher learning rates, the model converges to nearly the same training loss and validation performance values. Similarly, we have observed that while converge delays as

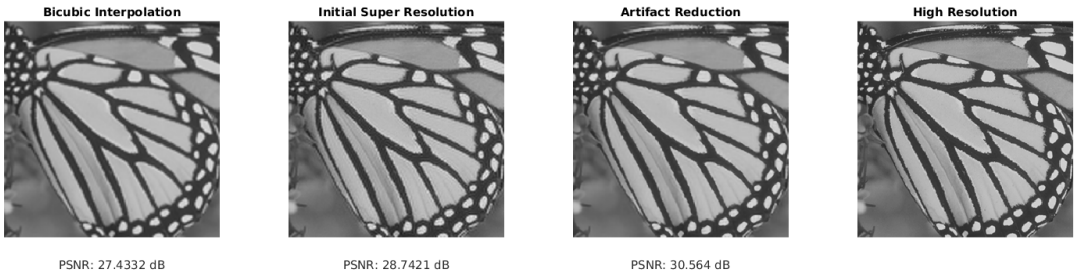


Figure 4: Intermediate Steps of Our Super Resolution System

Table I: Quantitative evaluation of state-of-the-art SR solutions (PSNR - SSIM score pairs).

Algorithm	Scale	Set5 [24]	Set14 [25]	BSDS100 [26]	Urban100 [27]
Bicubic	2	33.69 - 0.931	30.25 - 0.870	29.57 - 0.844	26.89 - 0.841
FNNSR [11]	2	35.20 - 0.943	31.40 - 0.895	30.58 - 0.877	-
Ours	2	**35.53 - 0.953**	**31.64 - 0.904**	**30.64 - 0.884**	**28.15 - 0.882**
RFL [3]	2	36.59 - 0.954	32.29 - 0.905	31.18 - 0.885	29.14 - 0.891
SelfExSR [27]	2	36.60 - 0.955	32.24 - 0.904	31.20 - 0.887	29.55 - 0.898
SRCNN [4]	2	36.72 - 0.955	32.51 - 0.908	31.38 - 0.889	29.53 - 0.896
FSRCNN [13]	2	37.05 - 0.956	32.66 - 0.909	31.53 - 0.892	29.88 - 0.902
VDSR [5]	2	37.53 - 0.959	33.05 - 0.913	31.90 - 0.896	30.77 - 0.914
LapSRN [6]	2	37.52 - 0.959	33.08 - 0.913	31.80 - 0.895	30.41 - 0.910
Bicubic	3	30.41 - 0.869	27.55 - 0.775	27.22 - 0.741	24.47 - 0.737
FNNSR [11]	3	31.42 - 0.883	28.32 - 0.802	27.79 - 0.772	-
Ours	3	**31.44 - 0.906**	**28.41 - 0.828**	**27.78 - 0.788**	**24.78 - 0.781**
RFL [3]	3	32.47 - 0.906	29.07 - 0.818	28.23 - 0.782	25.88 - 0.792
SelfExSR [27]	3	32.66 - 0.910	29.18 - 0.821	28.30 - 0.786	26.45 - 0.810
SRCNN [4]	3	32.78 - 0.909	29.32 - 0.823	28.42 - 0.788	26.25 - 0.801
FSRCNN [13]	3	33.18 - 0.914	29.37 - 0.824	28.53 - 0.791	26.43 - 0.808
VDSR [5]	3	33.67 - 0.921	29.78 - 0.832	28.83 - 0.799	27.14 - 0.829
LapSRN [6]	3	33.82 - 0.922	29.87 - 0.832	28.82 - 0.798	27.07 - 0.828
Bicubic	4	28.43 - 0.811	26.01 - 0.704	25.97 - 0.670	23.15 - 0.660
FNNSR [11]	4	29.35 - 0.827	26.62 - 0.727	26.42 - 0.696	-
Ours	4	**29.21 - 0.852**	**26.55 - 0.755**	**26.33 - 0.721**	**23.42 - 0.701**
RFL [3]	4	30.17 - 0.855	27.24 - 0.747	26.76 - 0.708	24.20 - 0.712
SelfExSR [27]	4	30.34 - 0.862	27.41 - 0.753	26.84 - 0.713	24.83 - 0.740
SRCNN [4]	4	30.50 - 0.863	27.52 - 0.753	26.91 - 0.712	24.53 - 0.725
FSRCNN [13]	4	30.72 - 0.866	27.61 - 0.755	26.98 - 0.715	24.62 - 0.728
VDSR [5]	4	31.35 - 0.883	28.02 - 0.768	27.29 - 0.726	25.18 - 0.754
LapSRN [6]	4	31.54 - 0.885	28.19 - 0.772	27.32 - 0.727	25.21 - 0.756

the number of trainable parameters gets larger, the model converges to nearly the same performance scores despite changes in the number of neurons hyper-parameter.

In the artifact reduction model (AR-CNN [23]) we use as a post-processing step, four different pre-trained models are available. Each model is trained over a different image set, constructed with different JPEG compression quality values ($Q \in \{10, 20, 30, 40\}$). We choose the best model for our post-SR processing purposes according to the PSNR values obtained on the validation set. The best PSNR value is obtained for $Q = 40$, which is the highest JPEG quality.

B. *Experimental results*

In this section, we present our main quantitative results with comparisons to contemporary deep SR approaches, analyze the effect of the artifact-reduction network in detail, give an ablative study and finally discuss our SR model through qualitative examples.

Main results. In Table I, we provide the PSNR and SSIM scores of our approach and a number of other SR approaches. The table consists of three sections, corresponding to respectively $2\times$, $3\times$ and $4\times$ scaling factors. The last four columns correspond to the performance scores obtained on the Set-5, Set-14, BSDS100, and, Urban100 datasets. For each method and each dataset value, we present the corresponding pair of PSNR - SSIM scores.

In the results shown in Table I, we observe that bicubic interpolation, which is the most basic method shown in the table and is also the first step of our SR approach, obtains the lower PSNR and SSIM results, as expected. The results of the bicubic interpolation can be seen as the baseline performance for all SR methods. We observe that our approach obtains significant improvements in PSNR and SSIM scores compared to the bicubic interpolation baseline. However, we also observe that our approach obtains relatively lower scores, especially in terms of the PSNR scores, compared to other methods, especially the spatial-domain SR techniques. This is not surprising to consider that most spatial domain SR methods directly aim to minimize the reconstruction loss, which in fact corresponds to optimizing the PSNR score. In fact, we observe that our method obtains much more competitive scores

in terms of the SSIM metric.

To improve our understanding of the DCT spectral representation for SR purposes, we repeat the identical mean square analysis in frequency domain which we made in Section III now using the SR method outputs. In the Figure 5a, 5b, 5c and 5d, we present the mean squared error analysis results for bicubic interpolation, our model, SRCNN and LapSRN, respectively. First of all, as previously discussed, we observe again that bicubic interpolation is most problematic for mid-frequency terms. While we observe significant error reductions made by the SRCNN and LapSRN models in the region between low and middle frequencies, we still observe rather large errors on the mid-frequency components. We also observe that our model is capable to correct the target problematic region coefficients considerably. However, we observe that the error reduction is not perfect, as there are still errors in relatively lower frequencies of the targeted region. Overall, we observe that for all SR methods in consideration, middle frequency DCT terms remain to be the most problematic region.

Analysis of artifact reduction. One important question that needs to be answered is the role of artifact reduction network on the SR performance scores we obtain. Therefore, in order to measure the significance of artifact reduction module, we evaluate its effect using different module combinations: using (i) only bicubic interpolation, (ii) bicubic interpolation followed by artifact reduction, and, (iii) bicubic interpolation followed by super-resolution and artifact reduction. For these three experiments, we obtain 33.69 dB, 33.82 dB and 35.53 dB PSNR scores on the Set5 dataset, respectively. These results show that even if artifact reduction solution is a deep image enhancement architecture, utilizing AR network alone improves the bicubic interpolation result by only 0.13 dB, which is far smaller than the improvement obtained by our complete approach. This result shows that the artifact reduction itself is not suitable for replacing the SR network.

Ablative study. In Table II, we present a detailed analysis on our design choices, particularly on applying artifact reduction as a post-processing step and operating only on a subset of DCT components. More specifically, the table shows results for the scaling factors $2\times$, $3\times$ and $4\times$, each one corresponding to one section. For each scaling factor, we present the PSNR and SSIM scores on three different datasets, for bicubic interpolation, our super-resolution model only, our super-resolution approach with artifact reduction and our super-resolution model operating at all 1024 DCT components.

From these results, we first see that, SR-only approach (without artifact reduction) yields significant improvements over the bicubic interpolation, consistently across the scaling factors and across the test datasets. We also see that artifact reduction provides valuable improvements over the SR model outputs. Finally, we observe that using all frequencies in super-resolution mapping yields results even below bicubic interpolation images. While the difficulty of training over all frequencies is a factor here, we have observed that poor prediction of the offset value plays a dominant role in poor performance in this case. Overall, these results confirm the design choices that we have made in our approach.

Qualitative results. In this section, we qualitatively evaluate our approach with comparisons to reference spatial-domain SR methods and ground truth HR images. For simplicity, we focus on the $3\times$ scaling factor and images from the Set14 dataset.

The sample images are shown in Figure 6. Each row shows the images obtained using SRCNN, LapSRN and our method, followed by the ground truth image. We observe that our method shows better results in capturing certain patterns in comparison to SRCNN and LapSRN, while yielding less sharp edges, as expected. For instance, in the second image, we can see that pattern on the tablecloth is reconstructed at better fidelity compared to the spatial domain methods. However, especially in the third example, it can be seen that our method tends to yield slightly more blurry images, as a result of focusing on improving mid-frequency DCT terms.

V. Conclusions

In this work, we have explored the use of spectral representations for developing a complete single-image super-resolution model. Our SR system involves of two main stages. In the first stage, we learn a super-resolution mapping from LR images to HR images entirely in the spectral domain, using DCT representation of the images. Here, we use the bicubic interpolation to initiate the output image and choose the subset of most problematic DCT frequency components at train time. Our second stage aims to remove ringing artifacts caused by spectral transformations used in the first phase, using an artifact reduction network.

Overall, our results show that, while deep SR methods formulated in the spatial domain yields better PSNR scores, there are cases where applying super-resolution in the spectral domain is advantageous, especially at reconstructing patterns. We also observe that there are specific DCT components for which the mainstream SR models yield highly erroneous estimates. These observations suggest that despite challenges of formulating super-resolution purely in the spectral domain, this line of research is promising in many aspects, and can be a way to enhance and improve, especially through developing hybrid models that jointly model the spectral and spatial domain. Additionally, towards obtaining better results, spectral super-resolution methods can potentially be enhanced through more sophisticated network designs and introducing adversarial training strategies.

References

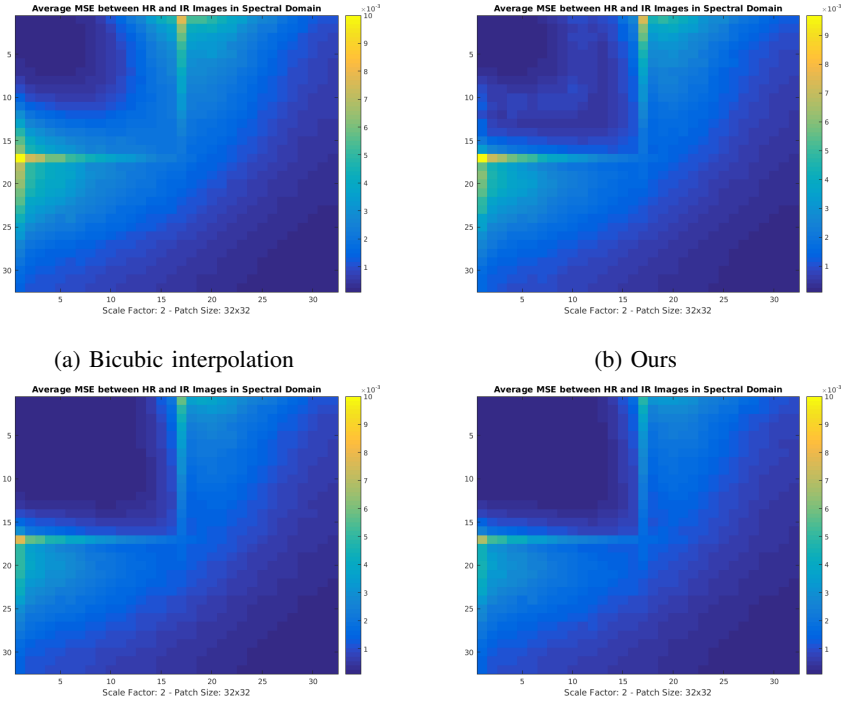
[1] R. Timofte, V. De Smet, and L. Van Gool, "A+: Adjusted anchored neighborhood regression for fast super-resolution," in *Asian Conference on Computer Vision*. Springer, 2014, pp. 111–126.

[2] J. Yang, J. Wright, T. S. Huang, and Y. Ma, "Image super-resolution via sparse representation," *IEEE International Conference on Image Processing*, vol. 19, no. 11, pp. 2861–2873, 2010.

[3] S. Schulter, C. Leistner, and H. Bischof, "Fast and accurate image upscaling with super-resolution forests," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2015, pp. 3791–3799.

[4] C. Dong, C. C. Loy, K. He, and X. Tang, "Image super-resolution using deep convolutional networks," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 38, no. 2, pp. 295–307, 2016.

[5] J. Kim, J. Kwon Lee, and K. Mu Lee, "Accurate image super-resolution using very deep convolutional networks," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2016, pp. 1646–1654.



(a) Bicubic interpolation (b) Ours

(c) SRCNN (d) LapSRN

Figure 5: Mean squared error of DCT coefficients between bicubic interpolated and HR patches at the scaling factor of $2\times$.

Table II: Quantitative evaluation for analyzing the effect of artifact reduction and operating on a subset of DCT components. PSNR-SSIM score pairs are shown.

Setup	Scale	Set5 [24]		Set14 [25]		BSDS100 [26]	
Bicubic	2	33.69	0.931	30.25	0.870	29.57	0.844
SR	2	34.72	0.941	31.15	0.903	30.37	0.889
SR+AR	2	35.53	0.953	31.64	0.904	30.64	0.884
SR - all frequencies	2	27.68	0.893	25.98	0.865	24.98	0.862
Bicubic	3	30.41	0.869	27.55	0.775	27.22	0.741
SR	3	31.08	0.901	28.17	0.826	27.66	0.801
SR+AR	3	31.44	0.906	28.41	0.828	27.78	0.788
SR - all frequencies	3	24.37	0.889	21.46	0.755	21.65	0.735
Bicubic	4	28.43	0.811	26.01	0.704	25.97	0.670
SR	4	28.93	0.846	26.39	0.753	26.19	0.719
SR+AR	4	29.21	0.852	26.55	0.755	26.33	0.721
SR - all frequencies	4	22.13	0.835	21.32	0.708	20.88	0.658

[6] W.-S. Lai, J.-B. Huang, N. Ahuja, and M.-H. Yang, "Deep laplacian pyramid networks for fast and accurate super-resolution," *arXiv preprint arXiv:1704.03915*, 2017.

[7] S. Anwar, S. Khan, and N. Barnes, "A deep journey into super-resolution: A survey," *arXiv preprint arXiv:1904.07523*, 2019.

[8] O. Rippel, J. Snoek, and R. P. Adams, "Spectral representations for convolutional neural networks," in *Advances in Neural Information Processing Systems*, 2015, pp. 2449–2457.

[9] Y. Wang, C. Xu, S. You, D. Tao, and C. Xu, "Cnnpack: Packing convolutional neural networks in the frequency domain," in *Advances in Neural Information Processing Systems*, 2016, pp. 253–261.

[10] N. Kumar, R. Verma, and A. Sethi, "Convolutional neural networks for wavelet domain super resolution," *Pattern Recognition Letters*, vol. 90, pp. 65–71, 2017.

[11] J. Li, S. You, and A. Robles-Kelly, "A frequency domain neural network for fast image super-resolution," in *International Joint Conference on Neural Networks*. IEEE, 2018, pp. 1–8.

[12] S. Xue, W. Qiu, F. Liu, and X. Jin, "Faster image super-resolution by improved frequency-domain neural networks," *Signal, Image and Video Processing*, pp. 1–9, 2019.

[13] C. Dong, C. C. Loy, and X. Tang, "Accelerating the super-resolution convolutional neural network," in *European Conference on Computer Vision*. Springer, 2016, pp. 391–407.

[14] C. Ledig, L. Theis, F. Huszár, J. Caballero, A. Cunningham, A. Acosta, A. Aitken, A. Tejani, J. Totz, Z. Wang *et al.*, "Photo-realistic single image super-resolution using a generative adversarial network," *arXiv preprint arXiv:1609.04802*, 2016.

[15] T. Dai, J. Cai, Y. Zhang, S.-T. Xia, and L. Zhang, "Second-order attention network for single image super-resolution," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2019, pp. 11 065–11 074.

[16] Y. Wang, F. Perazzi, B. McWilliams, A. Sorkine-Hornung, O. Sorkine-Hornung, and C. Schroers, "A fully progressive approach to single-image super-resolution," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 864–873.

[17] A. V. Oppenheim, *Discrete-time signal processing*. Pearson Education India, 1999.

[18] K. R. Rao and P. Yip, *Discrete cosine transform: algorithms, advantages, applications*. Academic press, 2014.

[19] R. Clarke, "Relation between the karhunen loeve and cosine transforms," in *IEE Proceedings F (Communications, Radar and Signal Processing)*, vol. 128, no. 6. IET, 1981, pp. 359–360.

[20] N. Srivastava, G. E. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, "Dropout: a simple way to prevent neural networks from overfitting." *Journal of Machine Learning Research*, vol. 15, no. 1,

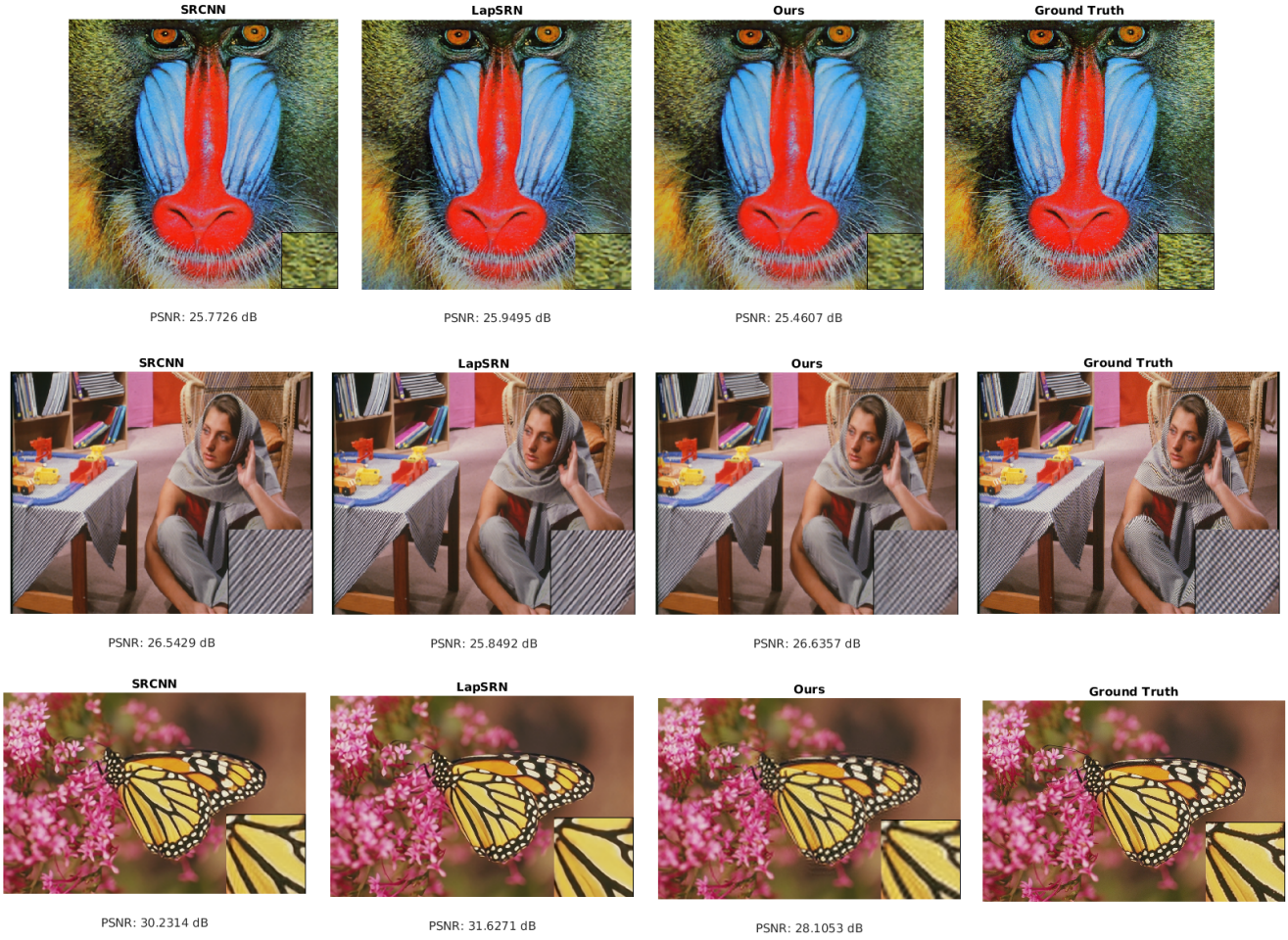


Figure 6: Example super-resolution results obtained using the SRCNN, LapSRN, our method with comparison to the ground truth high-resolution images. The images are taken from the Set14 dataset. The shown super-resolution results correspond to $3\times$ scaling factor.

pp. 1929–1958, 2014.

[21] X. Glorot and Y. Bengio, "Understanding the difficulty of training deep feedforward neural networks," in *International Conference on Artificial Intelligence and Statistics*, 2010, pp. 249–256.

[22] D. Kingma and J. Ba, "Adam: A method for stochastic optimization," *arXiv preprint arXiv:1412.6980*, 2014.

[23] C. Dong, Y. Deng, C. Change Loy, and X. Tang, "Compression artifacts reduction by a deep convolutional network," in *IEEE International Conference on Computer Vision*, 2015, pp. 576–584.

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

[25] W. Shi, J. Caballero, F. Huszár, J. Totz, A. P. Aitken, R. Bishop, D. Rueckert, and Z. Wang, "Real-time single image and video super-resolution using an efficient sub-pixel convolutional neural network," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2016, pp. 1874–1883.

[26] P. Arbelaez, M. Maire, C. Fowlkes, and J. Malik, "Contour detection and hierarchical image segmentation," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 33, no. 5, pp. 898–916, 2010.

[27] J.-B. Huang, A. Singh, and N. Ahuja, "Single image super-resolution from transformed self-exemplars," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2015, pp. 5197–5206.

[28] Zhou Wang, A. C. Bovik, H. R. Sheikh, and E. P. Simoncelli, "Image quality assessment: from error visibility to structural similarity," *IEEE Transactions on Image Processing*, vol. 13, no. 4, pp. 600–612, April 2004.

Onur Aydin Onur Aydn received the B.S. degree in Electrical and Electronics Engineering from Bilkent University, Turkey in 2014 and the M.S. degree in Computer Engineering from Bilkent University, Turkey in 2018. His research interests include machine learning and computer vision, with special interest in image enhancement and learning with weak supervision.

Ramazan Gokberk Cinbis graduated from Bilkent University, Turkey, in 2008, and received an M.A. degree from Boston University, USA, in 2010. He was a doctoral student at INRIA Grenoble, France, between 2010-2014, and received a PhD degree from Université de Grenoble, France, in 2014. He is currently an Assistant Professor at METU, Ankara, Turkey. His research interests include machine learning and computer vision, with special interest in deep learning with incomplete weak supervision.