Image Super-Resolution by Neural Texture Transfer

Zhifei Zhang
Adobe Research
zzhang@adobe.com

Zhaowen Wang
Adobe Research
zhawang@adobe.com

Zhe Lin
Adobe Research
zlin@adobe.com

Hairong Qi
University of Tennessee
hqi@utk.edu

Abstract

Due to the significant information loss in low-resolution (LR) images, it has become extremely challenging to further advance the state-of-the-art of single image super-resolution (SISR). Reference-based super-resolution (RefSR), on the other hand, has proven to be promising in recovering high-resolution (HR) details when a reference (Ref) image with similar content as that of the LR input is given. However, the quality of RefSR can degrade severely when Ref is less similar. This paper aims to unleash the potential of RefSR by leveraging more texture details from Ref images with stronger robustness even when irrelevant Ref images are provided. Inspired by the recent work on image stylization, we formulate the RefSR problem as neural texture transfer. We design an end-to-end deep model which enriches HR details by adaptively transferring the texture from Ref images according to their textural similarity. Instead of matching content in the raw pixel space as done by previous methods, our key contribution is a multi-level matching conducted in the neural space. This matching scheme facilitates multi-scale neural transfer that allows the model to benefit more from those semantically related Ref patches, and gracefully degrade to SISR performance on the least relevant Ref inputs. We build a benchmark dataset for the general research of RefSR, which contains Ref images paired with LR inputs with varying levels of similarity. Both quantitative and qualitative evaluations demonstrate the superiority of our method over state-of-the-art[1].

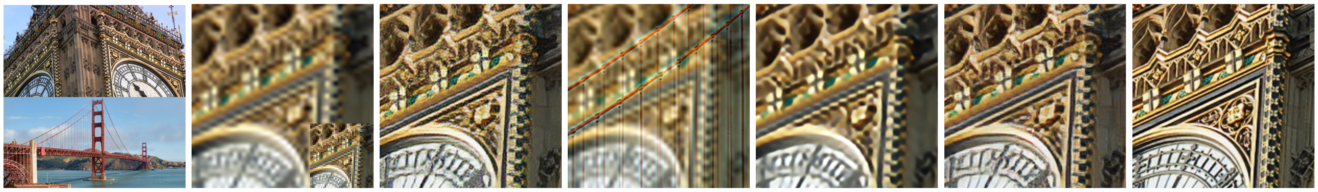
1. Introduction

The traditional single image super-resolution (SISR) problem is defined as recovering a high-resolution (HR) image from its low-resolution (LR) observation [38]. As in other fields of computer vision studies, the introduction of convolutional neural networks (CNNs) [5, 37, 22, 25, 35, 13] has greatly advanced the state-of-the-art of SISR. However, due to the ill-posed nature of SISR problems, most existing methods still suffer from blurry results at large upscaling factors, *e.g.*, 4×, especially when it comes to the recovery of fine texture present in the original HR image but lost in its LR counterpart. In recent years, perceptual-related constraints, *e.g.*, perception loss [20] and adversarial loss [11], have been introduced to the SISR problem formulation, leading to major breakthroughs on visual quality under large upscaling factors [24, 30]. However, they tend to hallucinate fake textures and even produce artifacts.

This paper diverts from the traditional SISR and explores the reference-based super-resolution (RefSR). RefSR utilizes rich textures from the HR references (Ref) to compensate for the lost details in the LR images, relaxing the ill-posed issue and producing more detailed and realistic textures with the help of reference images. Note that the Ref images can be obtained from various sources like photo albums, video frames, web image search, etc. There are existing RefSR approaches [8, 3, 7, 33, 39, 34, 27, 41] that adopt internal examples (self-example) or external high-frequency information to enhance textures. However, these approaches assume the reference images possess similar content as that of the LR image and/or with good alignment. Otherwise, their performance would significantly degrade and even become worse than SISR methods. In contrast, the Ref images play a different role in our setting: it does not require well alignment or similar content to the LR image. Instead, we only intend to transfer the semantically relevant texture from Ref images to the output SR image. Ideally, a robust RefSR algorithm should outperform SISR when good Ref images are given, and achieve comparable performance as SISR when Ref images are not provided or do not possess relevant texture at all. Note that content similarity would infer texture similarity but not vice versa.

Inspired by the recent work on image stylization [10, 20, 4], we propose a new RefSR algorithm, named Super-Resolution by Neural Texture Transfer (SRNTT), which *adaptively* transfers textures from the Ref images to the SR image. More specifically, SRNTT conducts local texture matching in the feature space and transfers matched textures to the final output through a deep model. The texture transfer model learns the complicated dependency between

[1] Code: https://github.com/ZZUTK/SRNTT



(a) Two Ref images (b) Bicubic & LR (c) SRGAN [24] (d) CrossNet [41] (L) (e) CrossNet [41] (U) (f) SRNTT (L) (g) SRNTT (U)

Figure 1: SRNTT (ours) is compared to SRGAN [24] (a state-of-the-art SISR method) and CrossNet [41] (a state-of-the-art RefSR method). (a) Two Ref images. The upper one (U) has similar content to the LR input as shown in (b) bottom-right corner, and the lower one (L) has distinct or unrelated content to the LR input. (c) Result of SRGAN. (d)(e) Results of CrossNet using two Ref images respectively. (f)(g) Results of SRNTT using two Ref images respectively.

LR and Ref textures, and leverages similar textures while suppressing dissimilar textures. The example in Fig. 1 illustrates the advantage of the proposed SRNTT compared with two state-of-the-art works, *i.e.*, SRGAN [24] (for SISR) and CrossNet [41] (for RefSR). SRNTT shows significant boost in synthesizing finer texture as compared to the other methods if using a Ref image with similar content (*i.e.*, Fig. 1(a) upper). Even using a Ref image with unrelated content (*i.e.*, Fig. 1(a) lower), SRNTT is still comparable to SRGAN (similar visual quality but less artifacts), demonstrating the adaptiveness/robustness of SRNTT to different Ref images of various levels of content similarity. By contrast, CrossNet would introduce undesired textures from the unrelated Ref image and shows severe performance degradation

In order to facilitate fair comparison and help advance research on the RefSR problem in general, we propose a new dataset, named CUFED5, which provides training and testing sets accompanied with references of different similarity levels in terms of content, texture, color, illumination, view point, etc. The main contributions of this paper are:

- We explore a more general RefSR problem, breaking the performance barrier in SISR (*i.e.*, lack of texture detail) and relaxing constraints in existing RefSR (*i.e.*, alignment assumption).
- We propose an end-to-end deep model, SRNTT, for the RefSR problem to recover the LR image conditioned on any given references by multi-scale neural texture transfer. We demonstrate the visual improvement, effectiveness, and adaptiveness of the proposed SRNTT by extensive empirical studies.
- We build a benchmark dataset, CUFED5, to facilitate the further research and performance evaluation of RefSR methods in handling references with different levels of similarity to the LR input image.

In the rest of this paper, we review the related works in Section 2. The network architecture and training criteria are discussed in Section 3. In Section 4, the proposed dataset CUFED5 is described in detail. The results of both quantitative and qualitative evaluations are presented in Section 5. Finally, Section 6 concludes this paper.

2. Related Works

2.1. Deep Learning based SISR

In recent years, deep learning based SISR has shown superior performance in terms of either PSNR or visual quality compared to non-deep-learning based methods [5, 37, 24]. The reader could refer to [29, 38] for more comprehensive review. Here we will only focus on deep learning based methods.

A milestone work that introduced CNN into SR was proposed by Dong et al. [5], where a three-layer fully convolutional network was trained to minimize the mean squared error (MSE) between the SR image and the original HR image. It demonstrated the effectiveness of deep learning in SR and achieved the state-of-the-art performance. Wang et al. [37] combined the strengths of sparse coding and deep network and made considerable improvement over previous models. To speed up the SR process, Dong et al. [6] and Shi et al. [31] extracted features directly from the LR image, that also achieved better performance compared to processing the upscaled LR image through bicubic interpolation. In recent years, the state-of-the-art performance (in PSNR) were all achieved by deep learning based models [22, 21, 25].

The above mentioned methods, in general, aim at minimizing MSE between the SR and HR images, which might not always be consistent with the human evaluation (*i.e.*, perceptual quality) [24, 30]. Therefore, perceptual-related constraints were incorporated to achieve better visual quality. Johnson et al. [20] demonstrated the effectiveness of adding perception loss using VGG [32]. Ledig et al. [24] introduced adversarial loss from the generative adversarial nets (GANs) [11] to minimize the perceptually relevant distance between the SR and HR images. Sajjadi et al. [30] further incorporated the texture matching loss based on the idea of style transfer [9, 10] to enhance the texture in the SR image. The proposed SRNTT is more closely related to [24, 30], where perceptual-related constraints (*i.e.*, perceptual loss and adversarial loss) are incorporated to recover more visually plausible SR images.

2.2. Reference-based Super-Resolution

In contrast to SISR where only a single LR image is used as input, RefSR methods introduce additional images to assist the SR process. In general, the reference images need to possess similar texture and/or content structure with the LR image. The references could be selected from adjacent frames in a video [26, 2], images from web retrieval [39], an external database (dictionary) [42], or images from different view points [41]. There is a batch of SR methods that refer to self patches/neighborhood [8, 3, 7, 16], which are widely known as self-example based SR. They do not utilize external references, thus more close to SISR problems. These works mostly build the mapping from LR to HR patches and fuse the HR patches at the pixel level or by a shallow model, which is insufficient to model the complicated dependency between the LR image and extracted details from the HR patches. A more generic scenario of utilizing the references was proposed by Yue et al. [39], which instantly retrieves similar images from web and conducts global registration and local matching. However, they made a strong assumption — the references have to be well aligned to the LR image. In addition, the shallow model for patch blending made its performance highly dependent on how well the references could be aligned. Zheng et al. [41] proposed a deep model based RefSR method and adopted optical flow to align input and reference. However, optical flow is limited in matching long distance correspondences, thus incapable of handling significantly misaligned references. The proposed SRNTT adopts the ideas of local texture (patch) matching which could handle long distance dependency. Like existing RefSR methods, we also "fuse" Ref texture to the final output, but we conduct it in the multi-scale feature space through a deep model, which enables the learning of complicated transfer process from references with scaling, rotation, or even non-rigid deformations.

3. Approach

The proposed SRNTT aims to estimate the SR image I^{SR} from its LR counterpart I^{LR} and the given reference images I^{Ref}, synthesizing plausible textures conditioned on I^{Ref} while preserving the consistency with I^{LR} in content. An overview of the proposed SRNTT is shown in Fig. 2. The main idea is to search for matching texture from I^{Ref} in the feature space and then transfer matched textures to I^{SR} in a multi-scale fashion, since the features are more robust to the variance of color and illumination. The multi-scale texture transfer simultaneously considers semantic (higher-level) and textual (lower-level) similarity between I^{LR} and I^{Ref}, leading to transferring related textures while suppressing irrelevant textures.

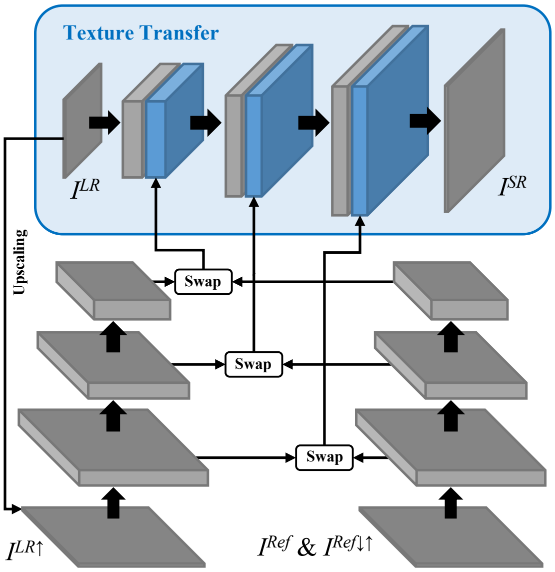


Figure 2: The proposed SRNTT framework with feature swapping and texture transfer.

In addition to minimizing the pixel and/or perceptual distance between the output I^{SR} and the original HR image I^{HR} as most existing SR methods do, we further regularize on the texture consistency between I^{SR} and the matched textures from I^{Ref}, enforcing the effectiveness of texture transfer. The final output I^{SR} is synthesized in an end-to-end manner. Texture searching and transfer will be discussed in Sections 3.1 and 3.2, respectively. Section 3.3 will detail the objective function of SRNTT.

3.1. Feature Swapping

We first conduct feature swapping which searches over the entire I^{Ref} for locally similar textures that can be used to replace (or swap) the texture features of I^{LR} for enhanced SR recovery. The feature searching is conducted in HR spatial coordinate to enable direct texture transfer to the final output I^{SR}. Following the self-example matching strategy [7], we first apply bicubic up-sampling on I^{LR} to get an upscaled LR image $I^{LR\uparrow}$ that has the same spatial size as I^{HR}. We also sequentially apply bicubic down-sampling and up-sampling with the same factor on I^{Ref} to obtain a blurry Ref image $I^{Ref\downarrow\uparrow}$ that matches the frequency band of $I^{LR\uparrow}$. Instead of estimating a global transformation or optical flow, we match the local patches in $I^{LR\uparrow}$ and $I^{Ref\downarrow\uparrow}$ so that there is no constraint on the global structure of the Ref image, which is a key advantage over CrossNet [41]. As LR and Ref patches may also differ in color and illumination, we match their similarity in the neural feature space $\phi(I)$ to emphasize the structural and textural information. We use inner product to measure the similarity between neural features:

$$s_{i,j} = \left\langle P_i(\phi(I^{LR\uparrow})), \frac{P_j(\phi(I^{Ref\downarrow\uparrow}))}{\|P_j(\phi(I^{Ref\downarrow\uparrow}))\|} \right\rangle, \tag{1}$$

where $P_i(\cdot)$ denotes sampling the i-th patch from neural feature map, and $s_{i,j}$ is the similarity between the i-th LR patch and the j-th Ref patch. The Ref patch feature is normalized for selecting the best match over all j. The similarity computation can be efficiently implemented as a set of convolution (or correlation) operations over all LR patches with each kernel corresponding to a Ref patch:

$$S_j = \phi(I^{LR\uparrow}) * \frac{P_j(\phi(I^{Ref\downarrow\uparrow}))}{\|P_j(\phi(I^{Ref\downarrow\uparrow}))\|}, \tag{2}$$

where S_j is the similarity map for the j-th Ref patch, and $*$ denotes the correlation operation. We use $S_j(x, y)$ to denote the similarity between the LR patch centered at location (x, y) and the j-th Ref patch. Both LR and Ref patches are densely sampled from their images. Based on the similarity score, we can construct a swapped feature map M to represent texture-enhanced LR image. Each patch in M centered at (x, y) is defined as

$$P_{\omega(x,y)}(M) = P_{j^*}(\phi(I^{Ref})), \;\; j^* = \arg\max_j S_j(x, y), \tag{3}$$

where $\omega(\cdot, \cdot)$ maps patch center to patch index. Note that while $I^{Ref\downarrow\uparrow}$ is used for matching (Eq. 2), the raw Ref I^{Ref} is used in swapping (Eq. 3) so that the HR information from the original references is preserved. Due to the dense sampling of LR patches, we take the average of the swapped features $P_{j^*}(\phi(I^{Ref}))$ in the regions where they overlap. The resulting swapped feature map M is used as the basis for the next texture transfer stage.

3.2. Neural Texture Transfer

Our texture transfer model is designed by merging multiple swapped texture feature maps into a base deep generative network at different feature layers corresponding to various scales, as illustrated in Fig. 2 (blue box). For each scale or neural layer l, a swapped feature map M_l is constructed using the method introduced above, with a texture feature encoder ϕ_l matching the current scale. The effectiveness of transferring texture across multiple layers is verified by the ablation study in Section 5.3.

We use residual blocks and skip connections [14, 15, 24] to build the base generative network. The network output ψ_l at layer l is defined recursively as

$$\psi_l = \left[\text{Res}\left(\psi_{l-1} \| M_{l-1}\right) + \psi_{l-1}\right] \uparrow_{2\times}, \tag{4}$$

where $\text{Res}(\cdot)$ denotes the residual blocks, $\|$ denotes channel-wise concatenation, and $\uparrow_{2\times}$ denotes $2\times$ upscaling with sub-pixel convolution [31]. The final SR result image is generated after L layers to reach target HR resolution:

$$I^{SR} = \text{Res}\left(\psi_{L-1} \| M_{L-1}\right) + \psi_{L-1} \tag{5}$$

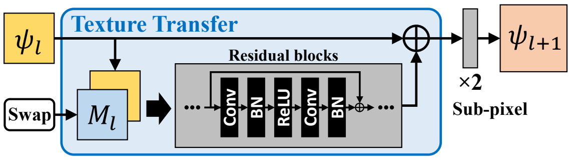


Figure 3: The network structure for texture transfer.

Fig. 3 illustrates the network structure of texture transfer at one scale, where the residual blocks extract related texture from M_l (*i.e.*, I^{Ref}) conditioned on ψ_l (*i.e.*, I^{LR}) and merge it with target content.

Different from traditional SISR methods that only reduce the difference between I^{SR} and the ground truth I^{HR}, our proposed SRNTT method further takes into account the texture difference between I^{SR} and I^{Ref}. That is, we require the texture of I^{SR} to be similar as the swapped feature map M_l in the feature space of ϕ_l. Specifically, we define a texture loss $\mathcal{L}_{tex}$ as

$$\mathcal{L}_{tex} = \sum_l \lambda_l \left\| Gr\left(\phi_l(I^{SR}) \cdot S_l^*\right) - Gr\left(M_l \cdot S_l^*\right) \right\|_F, \tag{6}$$

where $Gr(\cdot)$ computes the Gram matrix, and λ_l is a normalization factor corresponding to the feature size of layer l. S_l^* is a weighting map for all LR patches calculated as the best matching score in Eq. 3. Intuitively, textures dissimilar to I^{LR} will have lower weight, and thus receiving lower penalty in texture transfer. In this way, the texture transfer from I^{Ref} to I^{SR} is adaptively enforced based on the Ref image quality, leading to more robust texture hallucination as demonstrated in Section 5.3.

3.3. Training Objective

In order to 1) preserve the spatial structure of the LR image, 2) improve the visual quality of the SR image, and 3) take advantage of the rich texture from Ref images, our objective function combines reconstruction loss $\mathcal{L}_{rec}$, perceptual loss $\mathcal{L}_{per}$, adversarial loss $\mathcal{L}_{adv}$, and texture loss $\mathcal{L}_{tex}$. The reconstruction loss is adopted in most SR methods. The perceptual and adversarial losses improve visual quality. The texture loss already discussed in Eq. 6 is specific to RefSR.

Reconstruction loss aims to achieve higher PSNR, usually measured in terms of mean square error (MSE). In this paper, we adopt the ℓ_1-norm,

$$\mathcal{L}_{rec} = \left\| I^{HR} - I^{SR} \right\|_1, \tag{7}$$

The ℓ_1-norm would further sharpen I^{SR} as compared to MSE. In addition, it is consistent to the objective of WGAN-GP, which will be discussed later in the adversarial loss.

Perceptual loss has been investigated in recent SR works [1, 20, 24, 30] for better visual quality. We adopt the relu5_1 layer of VGG19 [32],

$$\mathcal{L}_{per} = \frac{1}{V} \sum_{i=1}^{C} \left\| \phi_i(I^{HR}) - \phi_i(I^{SR}) \right\|_F, \tag{8}$$

where V and C indicate the volume and channel number of the feature maps, respectively, and ϕ_i denotes the ith channel of the feature maps extracted from the hidden layer of VGG19 model. $\| \cdot \|_F$ denotes the Frobenius norm.

Adversarial loss could significantly enhance the sharpness/visual quality of synthesized images [19, 40]. Here, we adopt WGAN-GP [12], which improves upon WGAN by penalizing the gradient, achieving more stable results. Because the Wasserstein distance in WGAN is based on ℓ_1-norm, we use ℓ_1-norm as the reconstruction loss (Eq. 7). Intuitively, consistent objectives would facilitate the optimization process. The adversarial loss is expressed as

$$\mathcal{L}_{adv} = -\mathbb{E}_{\tilde{x}\sim\mathbb{P}_g}[D(\tilde{x})], \tag{9}$$

$$\min_G \max_{D\in\mathcal{D}} \mathbb{E}_{x\sim\mathbb{P}_r}[D(x)] - \mathbb{E}_{\tilde{x}\sim\mathbb{P}_g}[D(\tilde{x})], \tag{10}$$

where $\mathcal{D}$ is the set of 1-Lipschitz functions, and $\mathbb{P}_r$ and $\mathbb{P}_g$ are the model distribution and real distribution, respectively.

3.4. Implementation Details

We adopt a pre-trained VGG19 [32] model for feature swapping, which is well-known for its power of texture representation [9, 10]. Feature layers relu1_1, relu2_1, and relu3_1 are used as texture encoder ϕ_l's in multiple scales. To speed up the matching process, we only match on the relu3_1 layer and project the correspondence to layers relu2_1 and relu1_1, and use the same correspondence across all layers. The weights for $\mathcal{L}_{rec}$, $\mathcal{L}_{per}$, $\mathcal{L}_{adv}$, and $\mathcal{L}_{tex}$ are 1, 1e-4, 1e-6, and 1e-4, respectively. Adam optimizer is used with the learning rate of 1e-4. The network is pre-trained for 2 epochs, where only $\mathcal{L}_{rec}$ is applied. Then, all losses are involved to train another 20 epochs.

Our method can be easily extended to handle multiple Ref images. In all our RefSR experiments, we augment each I^{Ref} with its scaled and rotated versions to get more accurate texture matching results.

4. Dataset

For RefSR problems, the similarity between the LR and Ref images affects SR results significantly. In general, references with various levels of similarity to LR images should be provided for the purpose of both training and evaluating a RefSR algorithm. To the best of our knowledge, there has not been such a dataset available for public usage. We thus construct such a dataset with Ref images at various similarity levels based on the CUFED [36] dataset that contains 1,883 albums capturing diverse events in daily life. The size of each album varies between 30 and 100 images. Within each album, we collect image pairs in different similarity levels based on SIFT [28] feature matching, which characterizes local texture pattern that is in line with the objective of local texture matching.

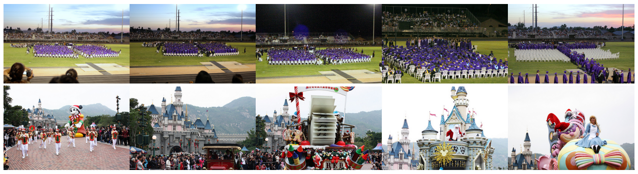


Figure 4: Examples from the CUFED5 testing set. From left to right are HR image and the corresponding Ref images of similarity levels L1, L2, L3 and L4, respectively.

We define four similarity levels from high to low, *i.e.*, L1, L2, L3, and L4, according to the number of best matches of SIFT features. From each paired images, we randomly crop 160×160 patches from one image as the original HR images, and the corresponding references are cropped from the other image. In this way, we collect 13,761 paired patches as the training set. For the testing dataset, each HR image is paired with all four levels of references in order to extensively evaluate the adaptiveness of a reference-based SR method. We use the similar way to collect image pairs as in building the training dataset. In total, the testing set contains 126 groups of samples. Each group consists of one HR image and four references at levels L1, L2, L3, and L4, respectively. Two examples from the testing set are shown in Fig. 4. We refer to the collected training and testing sets as CUFED5, which would largely facilitate the research on RefSR and provide a benchmark for fair comparison.

To evaluate the generalization capacity of the trained model on CUFED5, we test it on Sun80 [33] and Urban100 [16]. The Sun80 dataset has 80 natural images, each of which is accompanied by a series of web-searching references, while the Urban100 dataset contains building images without references.

5. Experimental Results

In this section, both quantitative and qualitative comparisons are conducted to demonstrate the advantages of the proposed SRNTT in terms of visual quality and texture enrichment. Following standard protocol, we obtain all LR images by bicubic downscaling (4×) from the HR images.

Table 1: PSNR/SSIM comparison of different SR methods on three datasets. Methods are grouped by SISR (top) and RefSR (bottom) with their respective best numbers in bold.

Algorithm	CUFED5	Sun80 [33]	Urban100 [17]
Bicubic	24.18 / 0.684	27.24 / 0.739	23.14 / 0.674
SRCNN [5]	25.33 / 0.745	28.26 / 0.781	24.41 / 0.738
SelfEx [16]	23.22 / 0.680	27.03 / 0.756	24.67 / 0.749
SCN [37]	25.45 / 0.743	27.93 / 0.786	24.52 / 0.741
DRCN [22]	25.26 / 0.734	27.84 / 0.785	25.14 / 0.760
LapSRN [23]	24.92 / 0.730	27.70 / 0.783	24.26 / 0.735
MDSR [25]	**25.93 / 0.777**	**28.52 / 0.792**	**25.51 / 0.783**
ENet [30]	24.24 / 0.695	26.24 / 0.702	23.63 / 0.711
SRGAN [24]	24.40 / 0.702	26.76 / 0.725	24.07 / 0.729
SRNTT-ℓ_2 (SISR)	25.91 / 0.776	28.46 / 0.790	25.50 / 0.783
Landmark [39]	24.91 / 0.718	27.68 / 0.776	—
CrossNet [41]	25.48 / 0.764	28.52 / 0.793	25.11 / 0.764
SRNTT-ℓ_2	**26.24 / 0.784**	**28.54 / 0.793**	**25.50 / 0.783**
SRNTT	25.61 / 0.764	27.59 / 0.756	25.09 / 0.774

5.1. Quantitative Evaluation

We compare the proposed SRNTT with the state-of-the-art SISR and RefSR algorithms[2] as shown in Table 1. The SISR methods in comparison are SRCNN [5], SelfEx [16], SCN [37], DRCN [22], LapSRN [23], MDSR [25], ENet [30], and SRGAN [24], among which MDSR [25] has achieved the state-of-the-art performance in PSNR in recent two years, while ENet [30] and SRGAN [24] are considered the state-of-the-art in visual quality. Two RefSR methods are also included in the comparison, *i.e.*, Landmark [39] and the recently proposed CrossNet [41], which outperforms previous RefSR methods.

For fair comparison, all learning-based methods are trained on the proposed CUFED5 dataset, and tested on CUFED5, Sun80 [33], and Urban100 [16], respectively. For fair comparison on PSNR/SSIM with those methods mainly minimizing MSE, *e.g.*, SCN and MDSR, we train a simplified version of SRNTT by only minimizing the MSE, *i.e.*, SRNTT-ℓ_2. Note that Table 1 shows the results of SRNTT-ℓ_2 in both SISR (upper block) and RefSR (lower block) settings. Specifically, the SRNTT-ℓ_2 under SISR setting uses the LR input as reference. In CUFED5 and Sun80 datasets, each input corresponds to multiple references, all of which are used in Landmark, SRNTT-ℓ_2 and SRNTT, while CrossNet uses the reference that yields the highest PSNR because CrossNet accepts only one reference.

In Table 1, SRNTT-ℓ_2 achieves the highest score on CUFED5 and Sun80 which have references, while performing comparably to MDSR (the highest score) on Urban100 which does not have references. Even with SISR setting on all datasets, SRNTT-ℓ_2 (SISR) performs similarly to the state-of-the-art. The proposed SRNTT, which uses adversarial loss that would increase visual quality but reduce PSNR, outperforms ENet and SRGAN in PSNR (even comparable to those methods that only minimize MSE), while at the same time achieving higher visual quality (finer texture and less artifacts) as shown by the examples in Fig. 5. A more comprehensive evaluation on visual quality will be conducted in Section 5.2. As demonstrated by the examples, SRNTT outperforms CrossNet in recovering fine texture from references. The main reason is that the references present large disparity/misalignment from the LR image, which CrossNet is incapable of handling.

Without loss of generality, examples from Sun80 and Urban100 are displayed in Fig. 5. With the help of references, SRNTT outperforms other SR methods on Sun80. On Urban100, however, there is no HR references. We use LR input as the reference and achieve finer texture that could be transferred from the LR image. In general, SRNTT would outperform existing SR methods with the assistance of references, and we could still achieve state-of-the-art SISR performance when there is no HR information from references. Section 5.3 will further demonstrate the adaptiveness of SRNTT by analyzing the performance on references of different similarity levels.

5.2. Qualitative Evaluation by User Study

To evaluate the visual quality of the SR images, we conduct user study, where SRNTT is compared to SCN [37], DRCN [22], MDSR [25], ENet [30], SRGAN [24], Landmark [39], and CrossNet [41]. We present the users with pair-wise comparisons, *i.e.*, SRNTT vs. other, and ask the users to select the one with higher resolution. For each reference level, 2,400 votes are collected on the testing results from the CUFED5 dataset. Fig. 6 shows the voting results, where the percentages favoring SRNTT denotes the percentage of users that prefer SRNTT as compared to the algorithms denoted along the horizontal axis. Overall, SRNTT significantly outperforms the other algorithms with over 90% users voting for SRNTT.

5.3. Ablation Studies

5.3.1 Effect of reference similarity

Similarity between LR and Ref images is a key factor to the success of RefSR methods. This section investigates the performance of CrossNet [41] and the proposed SRNTT at different reference levels. Table 2 lists the results at six levels of references, where "HR (warp)" denotes the reference obtained by random translation (quarter to half width/height), rotation (10∼30 degree), and scaling (1.2×∼2.0× upscaling) from the original HR image. L1,

[2] Implementation of SR algorithms in comparison:
SRCNN: http://mmlab.ie.cuhk.edu.hk/projects/SRCNN.html
SelfEx: https://sites.google.com/site/jbhuang0604/publications/struct_sr
SCN: http://www.ifp.illinois.edu/~dingliu2/iccv15/
DRCN: http://cv.snu.ac.kr/research/DRCN/
LapSRN: http://vllab.ucmerced.edu/wlai24/LapSRN/
MDSR: https://github.com/LimBee/NTIRE2017
ENet: https://webdav.tue.mpg.de/pixel/enhancenet/
SRGAN: https://github.com/tensorlayer/srgan
CrossNet: https://github.com/htzheng/ECCV2018_CrossNet_RefSR

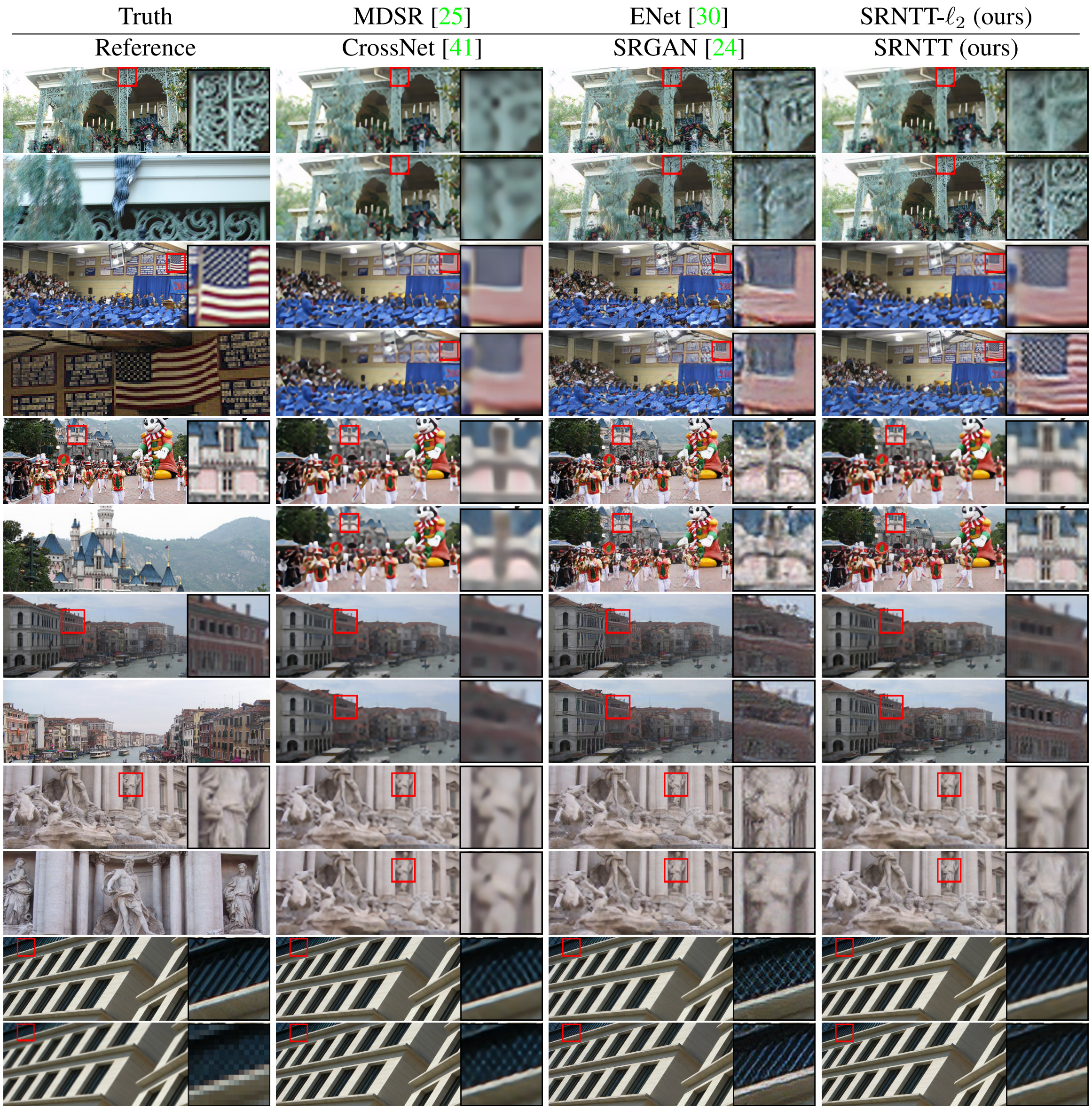


Figure 5: Visual comparison among different SR methods on CUFED5 (top three examples), Sun80 [33] (the forth and fifth examples), and Urban100 [16] (the bottom example whose reference image is the LR input).

L2, L3, and L4 are the four levels of references from the proposed CUFED5 dataset. "LR" means using the LR input image as the references (there is no external references). As compared to CrossNet, the SRNTT-ℓ_2 shows superior results at each reference level. At the "HR" level, SRNTT-ℓ_2 achieves significant improvement, which demonstrates the advantage of patch-wise matching over the alignment using optical flow. Comparing SRNTT and SRNTT-ℓ_2, SRNTT shows even higher PSNR at "HR" level but lower at other levels. This phenomenon emphasizes the effectiveness of texture loss in recovering fine textures when given highly similar references.

To further investigate the gap between the CrossNet and SRNTT, we conduct an experiment by replacing feature swapping with optical flow (FlowNet2 [18]) in the SRNTT framework. As shown in Table 2, "SRNTT-flow" shows large degradation even at "HR" level as compared to SRNTT, reflecting the limitation of optical flow in handling large disparity/misalignment. As the reference similarity level decreases, PSNR/SSIM of SRNTT reduces gracefully as well. At "LR" level, SRNTT still achieves comparable performance as the state-of-the-art SISR algorithms (Table 1). We observe that the PSNR of SRNTT-flow is higher than that of SRNTT at the "LR" level because the Ref is

Table 2: PSNR/SSIM at different reference levels on CUFED5 dataset. PM indicates if patch-based matching is used; GAN indicates if GAN and other perceptual losses are used.

	PM	GAN	HR (warp)	L1	L2	L3	L4	LR
CrossNet [41]			25.49 / .764	25.48 / .764	25.48 / .764	25.47 / .763	25.46 / .763	25.46 / .763
SRNTT-ℓ_2	✓		29.29 / .889	**26.15 / .781**	**26.04 / .776**	**25.98 / .775**	**25.95 / .774**	**25.91 / .776**
SRNTT-flow		✓	25.82 / .801	24.64 / .743	24.22 / .723	24.15 / .719	24.05 / .714	25.50 / .756
SRNTT	✓	✓	**33.87 / .959**	25.42 / .758	25.32 / .752	25.24 / .751	25.23 / .750	25.10 / .750

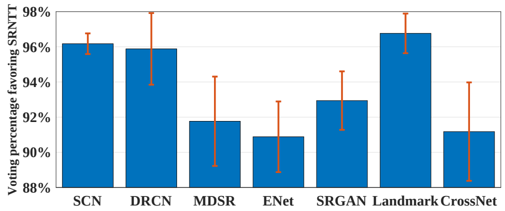


Figure 6: The user study result. SRNTT is compared to each algorithm along the horizontal axis, and the blue bars indicate the percentage of users favoring SRNTT results.

identical to the LR input. In this case, optical flow would easily align Ref to LR, while patch matching may have missed some matches.

5.3.2 Layers for feature swapping

As discussed in Section 3, feature swapping and transfer at multiple scales would increase the performance of SRNTT. Table 3 demonstrates the effectiveness of utilizing multiple scales as compared to using single scale. The relu1/2/3 denotes three layers/scales, *i.e.*, relu1_1, relu2_1, and relu3_1 from VGG19, used in SRNTT for feature swapping. We observe that the performance in PSNR decreases as reducing the number of scales. The relu3 gets the lowest PSNR because relu3_1 is a higher-level layer that carries less high-frequency information, contributing less to texture transfer as compared to relu1_1 and relu2_1. For each reference level, the PSNR follows the similar trend as the number of scales increases. However, it is interesting that relu3 shows decreasing and then increasing trend as the reference similarity decreases. This demonstrates the stronger adaptiveness of relu3 in preserving spacial structure, *i.e.*, low-similarity textures from the references are suppressed, and it tends to focus more on spacial reconstruction instead of textural recovery. Therefore, the multi-scale texture transfer using deep model gains extreme momentum on adaptively learning the complicated transfer process between the content and external texture.

5.3.3 Effect of texture loss

The weighted texture loss used in the proposed SRNTT is a key difference from most SR methods. Unlike those style transfer works, where the content image is significantly modified to carry the texture from the style image (*i.e.*, the reference), the proposed SRNTT avoids such "stylization" by local matching, adaptive neural transfer, and spatial/perceptual regularization. The local matching ensures spatially consistent texture, neural transfer gains adaptiveness on texture transfer, and spatial/perceptual regularization forces the spacial consistency globally. The effect of texture loss is shown in Fig. 7. The PSNR tested on CUFED5 are 25.25 and 25.61 for SRNTT w/o and with the texture loss, respectively. Without the texture loss, the finer texture from the references cannot be effectively transferred into the output.

Table 3: PSNR of using different VGG layers for feature swapping on different reference levels.

Layer	relu1	relu2	relu3	relu1/2	relu1/2/3
HR	28.39	28.66	24.83	30.39	33.87
L1	24.76	24.91	24.48	25.05	25.42
L2	24.68	24.86	24.22	25.00	25.32
L3	24.64	24.80	24.39	24.94	25.24
L4	24.63	24.79	24.45	24.92	25.23

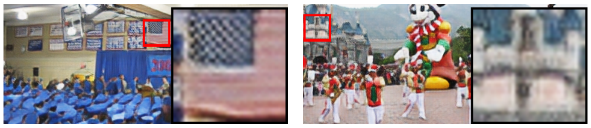


Figure 7: SR results with texture loss disabled have degraded quality compared with the same examples in Fig. 5.

6. Conclusion

This paper exploited the more generic RefSR problem where the references can be arbitrary images. We proposed SRNTT, an end-to-end network structure that performs multi-level adaptive texture transfer from the references to recover more plausible texture in the SR image. Both quantitative and qualitative experiments were conducted to demonstrate the effectiveness and adaptiveness of SRNTT. In addition, a new dataset CUFED5 was constructed to facilitate the evaluation of RefSR methods. It also provides a benchmark for future RefSR research.

References

[1] J. Bruna, P. Sprechmann, and Y. LeCun. Super-resolution with deep convolutional sufficient statistics. In *International Conference on Learning Representations (ICLR)*, 2016. 5

[2] J. Caballero, C. Ledig, A. Aitken, A. Acosta, J. Totz, Z. Wang, and W. Shi. Real-time video super-resolution with spatio-temporal networks and motion compensation. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017. 3

[3] H. Chang, D.-Y. Yeung, and Y. Xiong. Super-resolution through neighbor embedding. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2004. 1, 3

[4] T. Q. Chen and M. Schmidt. Fast patch-based style transfer of arbitrary style. In *Workshop in Constructive Machine Learning*. Advances in Neural Information Processing Systems, 2016. 1

[5] C. Dong, C. C. Loy, K. He, and X. Tang. Learning a deep convolutional network for image super-resolution. In *European Conference on Computer Vision (ECCV)*, 2014. 1, 2, 6

[6] C. Dong, C. C. Loy, and X. Tang. Accelerating the super-resolution convolutional neural network. In *European Conference on Computer Vision (ECCV)*, 2016. 2

[7] G. Freedman and R. Fattal. Image and video upscaling from local self-examples. *ACM Transactions on Graphics*, 30(2):12, 2011. 1, 3

[8] W. T. Freeman, T. R. Jones, and E. C. Pasztor. Example-based super-resolution. *IEEE Computer graphics and Applications*, 22(2):56–65, 2002. 1, 3

[9] L. Gatys, A. S. Ecker, and M. Bethge. Texture synthesis using convolutional neural networks. In *Advances in Neural Information Processing Systems*, 2015. 3, 5

[10] L. A. Gatys, A. S. Ecker, and M. Bethge. Image style transfer using convolutional neural networks. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016. 1, 3, 5

[11] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio. Generative adversarial nets. In *Advances in neural information processing systems*, 2014. 1, 2

[12] I. Gulrajani, F. Ahmed, M. Arjovsky, V. Dumoulin, and A. C. Courville. Improved training of Wasserstein GANs. In *Advances in Neural Information Processing Systems*, 2017. 5

[13] W. Han, S. Chang, D. Liu, M. Yu, M. Witbrock, and T. S. Huang. Image super-resolution via dual-state recurrent networks. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018. 1

[14] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016. 4

[15] K. He, X. Zhang, S. Ren, and J. Sun. Identity mappings in deep residual networks. In *European Conference on Computer Vision (ECCV)*, 2016. 4

[16] 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 (CVPR)*, 2015. 3, 5, 6, 7

[17] 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 (CVPR)*, 2015. 6

[18] E. Ilg, N. Mayer, T. Saikia, M. Keuper, A. Dosovitskiy, and T. Brox. Flownet 2.0: Evolution of optical flow estimation with deep networks. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017. 7

[19] P. Isola, J.-Y. Zhu, T. Zhou, and A. A. Efros. Image-to-image translation with conditional adversarial networks. In *IEEE conference on computer vision and pattern recognition*, 2017. 5

[20] J. Johnson, A. Alahi, and L. Fei-Fei. Perceptual losses for real-time style transfer and super-resolution. In *European Conference on Computer Vision (ECCV)*, 2016. 1, 2, 5

[21] 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 (CVPR)*, 2016. 2

[22] J. Kim, J. Kwon Lee, and K. Mu Lee. Deeply-recursive convolutional network for image super-resolution. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016. 1, 2, 6, 11

[23] W.-S. Lai, J.-B. Huang, N. Ahuja, and M.-H. Yang. Deep Laplacian pyramid networks for fast and accurate super-resolution. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017. 6

[24] C. Ledig, L. Theis, F. Huszar, 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. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017. 1, 2, 3, 4, 5, 6, 7, 11, 12

[25] B. Lim, S. Son, H. Kim, S. Nah, and K. M. Lee. Enhanced deep residual networks for single image super-resolution. In *The IEEE Conference on Computer*

Vision and Pattern Recognition (CVPR) Workshops, 2017. 1, 2, 6, 7, 11, 12

[26] C. Liu and D. Sun. A Bayesian approach to adaptive video super resolution. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2011. 3

[27] J. Liu, W. Yang, X. Zhang, and Z. Guo. Retrieval compensated group structured sparsity for image super-resolution. *IEEE Transactions on Multimedia*, 19(2):302–316, 2017. 1

[28] D. G. Lowe. Object recognition from local scale-invariant features. In *IEEE International Conference on Computer Vision (ICCV)*, 1999. 5

[29] K. Nasrollahi and T. B. Moeslund. Super-resolution: a comprehensive survey. *Machine Vision and Applications*, 25(6):1423–1468, 2014. 2

[30] M. S. Sajjadi, B. Scholkopf, and M. Hirsch. EnhanceNet: Single image super-resolution through automated texture synthesis. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017. 1, 2, 3, 5, 6, 7, 11, 12

[31] 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 (CVPR)*, 2016. 2, 4

[32] K. Simonyan and A. Zisserman. Very deep convolutional networks for large-scale image recognition. In *International Conference on Learning Representations (ICLR)*, 2015. 2, 5, 11

[33] L. Sun and J. Hays. Super-resolution from internet-scale scene matching. In *IEEE International Conference on Computational Photography (ICCP)*, 2012. 1, 5, 6, 7

[34] R. Timofte, V. De, and L. Van Gool. Anchored neighborhood regression for fast example-based super-resolution. In *IEEE International Conference on Computer Vision (ICCV)*, 2013. 1

[35] X. Wang, K. Yu, C. Dong, and C. Change Loy. Recovering realistic texture in image super-resolution by deep spatial feature transform. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018. 1

[36] Y. Wang, Z. Lin, X. Shen, R. Mech, G. Miller, and G. W. Cottrell. Event-specific image importance. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016. 5

[37] Z. Wang, D. Liu, J. Yang, W. Han, and T. Huang. Deep networks for image super-resolution with sparse prior. In *IEEE International Conference on Computer Vision (ICCV)*, 2015. 1, 2, 6, 11

[38] C.-Y. Yang, C. Ma, and M.-H. Yang. Single-image super-resolution: A benchmark. In *European Conference on Computer Vision (ECCV)*, 2014. 1, 2

[39] H. Yue, X. Sun, J. Yang, and F. Wu. Landmark image super-resolution by retrieving web images. *IEEE Transactions on Image Processing*, 22(12):4865–4878, 2013. 1, 3, 6, 11

[40] Z. Zhang, Y. Song, and H. Qi. Age progression/regression by conditional adversarial autoencoder. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017. 5

[41] H. Zheng, M. Ji, H. Wang, Y. Liu, and L. Fang. CrossNet: An end-to-end reference-based super resolution network using cross-scale warping. In *European Conference on Computer Vision (ECCV)*, 2018. 1, 2, 3, 6, 7, 8, 11, 12

[42] Y. Zhu, Y. Zhang, and A. L. Yuille. Single image super-resolution using deformable patches. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2014. 3

[43] Z. Zhu, F. Guo, H. Yu, and C. Chen. Fast single image super-resolution via self-example learning and sparse representation. *IEEE Transactions on Multimedia*, 16(8):2178–2190, 2014. 11

Supplementary

This supplementary further details the network structure of the proposed SRNTT and provides more visual comparisons between SRNTT and the other methods cited in the experimental results of the original paper. In addition, examples of using references from web search are provided to further illustrate the generation capacity of SRNTT. Also included here is the evaluation of SRNTT in terms of run time and effect of adopting different upscaling method in the feature swapping stage.

A. Network structure

As discussed in the section of "Approach" in the original paper, the proposed SRNTT has mainly two components: 1) feature swapping and 2) neural texture transfer. The feature swapping uses the pre-trained VGG19 model [32]. The elaboration in this section mainly focuses on the network structure of the neural texture transfer, as well as the discriminator. Tables A.1 and A.2 list the layer details of the neural texture transfer and the discriminator.

B. Run Time of SRNTT

The patch matching in the feature swapping is the most time-consuming part, which could take over 95% of the total run time. Table B.1 lists the run time of SRNTT and other related works, where all methods are tested on a $83 \times 125 \times 3$ LR input image with the upscaling factor of $4\times$. We run on a Quadro P5000 GPU. The "Forward" denotes the time of forwarding through the network, "Patch Matching" indicates the time cost by the patch-matching process (*i.e.*, obtaining the M_l that is concatenated to the main flow). For SRNTT, the total run time should be $0.162 + 4.865 = 5.027$ seconds, where the patch matching takes about 96% of the total run time. It is well known that patch matching is time-consuming, and its run time would increase quadratically with the size of the LR input and Ref images. Regardless of the patch-matching part, the run time of SRNTT is still comparable to SISR methods in the forward time. Note that CrossNet uses MDSR for pre-upscaling, so the forward time has included the run time of MDSR.

C. Pre-Upscaling in Feature Swapping

In the feature swapping of the proposed SRNTT (Section 3.1 in the original paper), the input LR image I^{LR} needs to be pre-upscaled to match the scale of I^{SR}. The upscaling method of obtaining $I^{LR\uparrow}$, as well as $I^{Ref\downarrow\uparrow}$, is optional, *e.g.*, bicubic interpolation, MDSR [25], SRGAN [24], etc. We experimented with alternative pre-upscaling methods, *i.e.*, MDSR and SRGAN, and they dont make big difference. Specifically, we use MDSR and SRGAN in pre-upscaling, and their PSNR/SSIM on CUFED5 dataset are $25.63/0.765$ and $25.61/0.763$, respectively. By contrast, using bicubic interpolation for pre-upscaling would achieve $25.61/0.764$.

Table A.1: The network structure of neural texture transfer, and the kernel size is 3×3 except for the last layer, *i.e.*, #63 uses 1×1 kernel.

#	**Layer name(s)**	**Output size**
0	Input	$H \times W \times 3$
1	Conv, ReLU	$H \times W \times 64$
2∼17	Residual blocks (Conv, BN, ReLU, Conv, BN)	$H \times W \times 64$
18	Conv, BN	$H \times W \times 64$
19	#1 + #18	$H \times W \times 64$
20	Concat: #19 $\Vert M_l$	$H \times W \times 320$
21	Conv, ReLU	$H \times W \times 64$
22∼37	Residual blocks (Conv, BN, ReLU, Conv, BN)	$H \times W \times 64$
38	Conv, BN	$H \times W \times 64$
39	#19 + #38	$H \times W \times 64$
40	Conv, SubPixel, ReLU	$2H \times 2W \times 64$
41	Concat: #40 $\Vert M_{l+1}$	$2H \times 2W \times 192$
42	Conv, ReLU	$2H \times 2W \times 64$
43∼50	Residual blocks (Conv, BN, ReLU, Conv, BN)	$2H \times 2W \times 64$
51	Conv, BN	$2H \times 2W \times 64$
52	#40 + #51	$2H \times 2W \times 64$
53	Conv, SubPixel, ReLU	$4H \times 4W \times 64$
54	Concat: #53 $\Vert M_{l+2}$	$4H \times 4W \times 128$
55	Conv, ReLU	$4H \times 4W \times 64$
56∼59	Residual blocks (Conv, BN, ReLU, Conv, BN)	$4H \times 4W \times 64$
60	Conv, BN	$4H \times 4W \times 64$
61	#53 + #60	$4H \times 4W \times 64$
62	Conv	$4H \times 4W \times 32$
63	Conv, tanh	$4H \times 4W \times 3$

Therefore, the performance of SRNTT is insensitive to the variance of pre-upscaling methods although these methods themselves show large diversity in SR performance, *i.e.*, bicubic interpolation gives blurry outputs, MDSR yields clean SR images, and SRGAN generates sharp results but with more artifacts.

D. Comparison to Other Algorithms

This section visually compares the methods cited in the experimental part of the original paper, namely, LapSRN [43], SCN [37], DRCN [22], MDSR [25], ENet [30], SRGAN [24], Landmark [39], CrossNet [41] and the proposed SRNTT. Figs. D.1–D.5 show examples from the

Table A.2: The network structure of the discriminator, kernel size is 3×3. The output size is scaled down by stride 2, and the parameter of LReLU is 0.2. In the training dataset, the original image size is 160×160.

#	Layer name(s)	Output size
0	Input	$160 \times 160 \times 3$
1	Conv, BN, LReLU	$160 \times 160 \times 32$
2	Conv, BN, LReLU	$80 \times 80 \times 32$
3	Conv, BN, LReLU	$80 \times 80 \times 64$
4	Conv, BN, LReLU	$40 \times 40 \times 64$
5	Conv, BN, LReLU	$40 \times 40 \times 128$
6	Conv, BN, LReLU	$20 \times 20 \times 128$
7	Conv, BN, LReLU	$20 \times 20 \times 256$
8	Conv, BN, LReLU	$10 \times 10 \times 256$
9	Conv, BN, LReLU	$10 \times 10 \times 512$
10	Conv, BN, LReLU	$5 \times 5 \times 512$
11	Flatten	12800
12	FC, LReLU	1024
13	FC	1

Table B.1: Run time of different SR algorithms on an LR input image with the size of $83 \times 125 \times 3$ ($4\times$ upscaling).

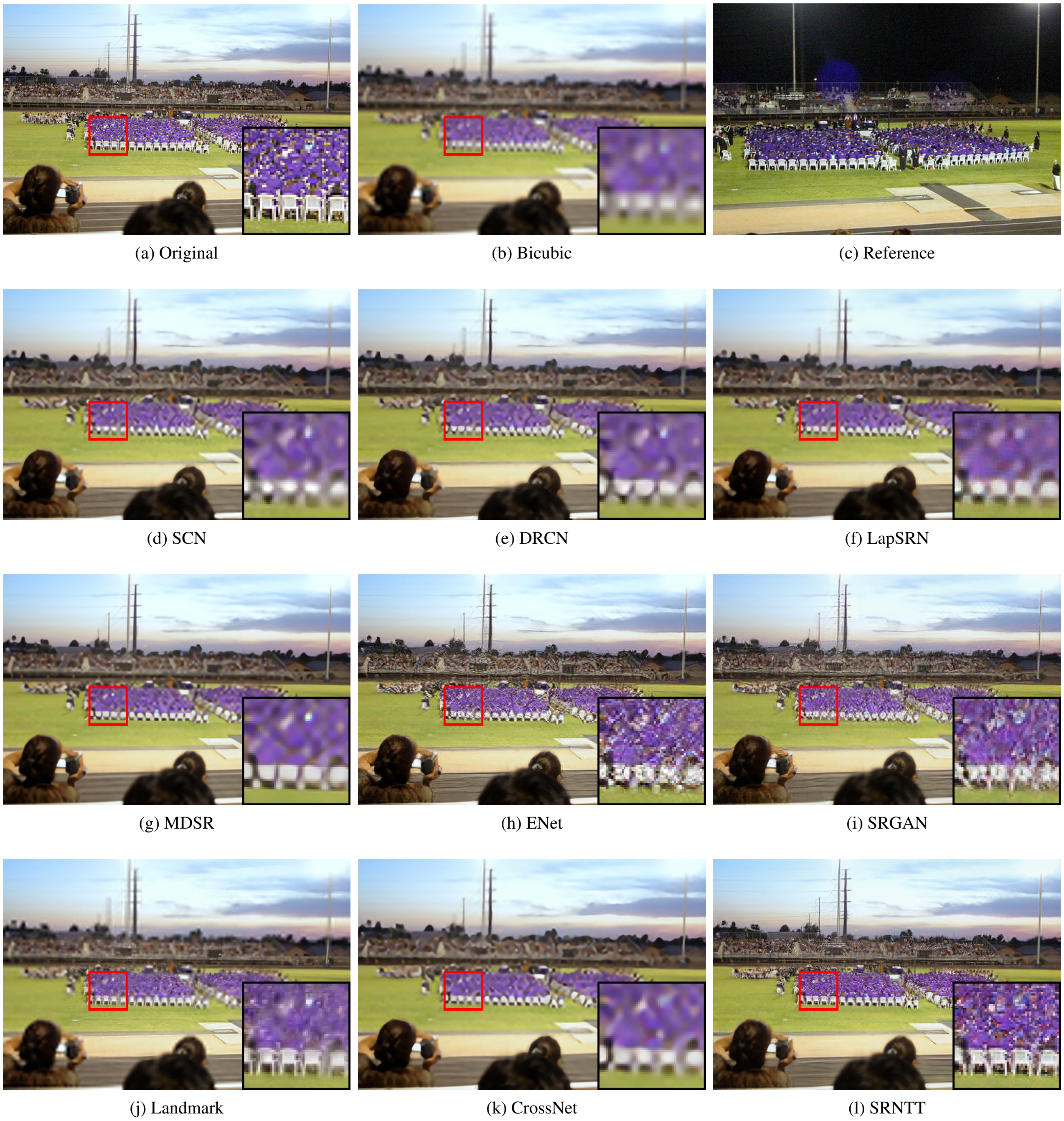
Algorithm	Forward (sec)	Patch Matching (sec)
MDSR [25]	0.684	—
ENet [30]	0.113	—
SRGAN [24]	0.115	—
CrossNet [41]	0.820	—
SRNTT (ours)	0.162	4.865

CUFED5 dataset. We observe that, in general, GAN-based methods, *i.e.*, ENet, SRGAN, and the proposed SRNTT, show more clear or less blurry SR results than MSE-based methods. In addition, compared to all the other methods, SRNTT presents more rich texture that is transferred from the reference.

E. References from Web Search

To further demonstrate the generation capacity of the proposed SRNTT, Figs. E.1–E.3 show examples that are easy to obtain relevant reference images from web search, *i.e.*, the famous landmark buildings — the Golden Gate Bridge and Louvre Museum, and the celebrity — Barack Obama. SRNTT is compared with the state-of-the-art SISR and RefSR methods. Specifically, MDSR [25] achieves the highest PSNR among existing SISR methods. ENet [30] and SRGAN [24] are comparable in visual quality. CrossNet [41] outperforms previous RefSR methods.

As compared to these algorithms, the proposed SRNTT presents more rich texture that is transferred from the reference. In addition, SRNTT shows sharper results but with less artifacts as compared to the GAN-based methods, *i.e.*, ENet and SRGAN. Meanwhile, SRNTT significantly outperforms the MSE-based methods, *i.e.*, MDSR and CrossNet, in enhancing finer texture and preserving comparable PSNR.



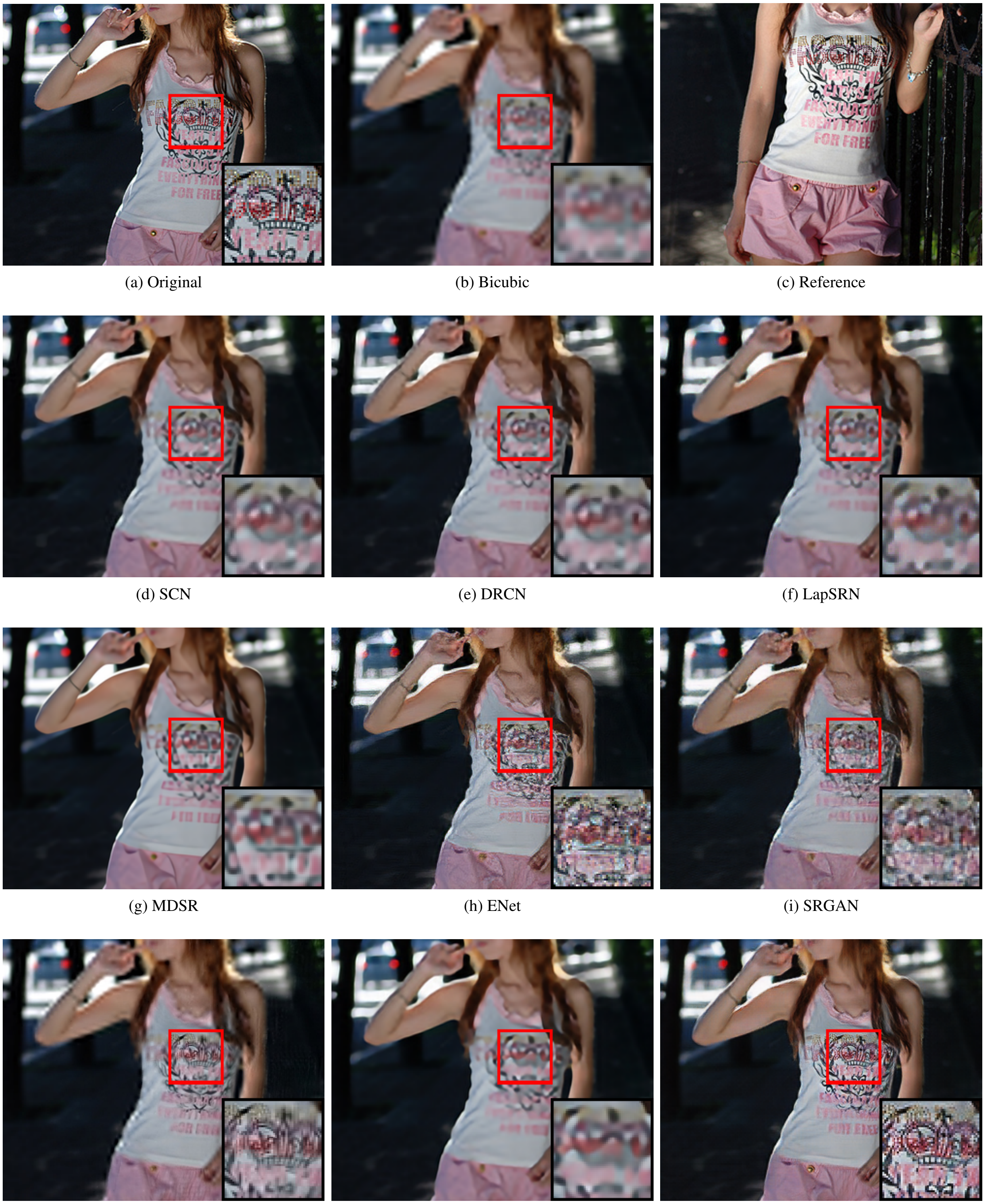
(a) Original (b) Bicubic (c) Reference

(d) SCN (e) DRCN (f) LapSRN

(g) MDSR (h) ENet (i) SRGAN

(j) Landmark (k) CrossNet (l) SRNTT

Figure D.1: Comparison of different SR algorithms on CUFED5 dataset.



(a) Original

(b) Bicubic

(c) Reference

(d) SCN

(e) DRCN

(f) LapSRN

(g) MDSR

(h) ENet

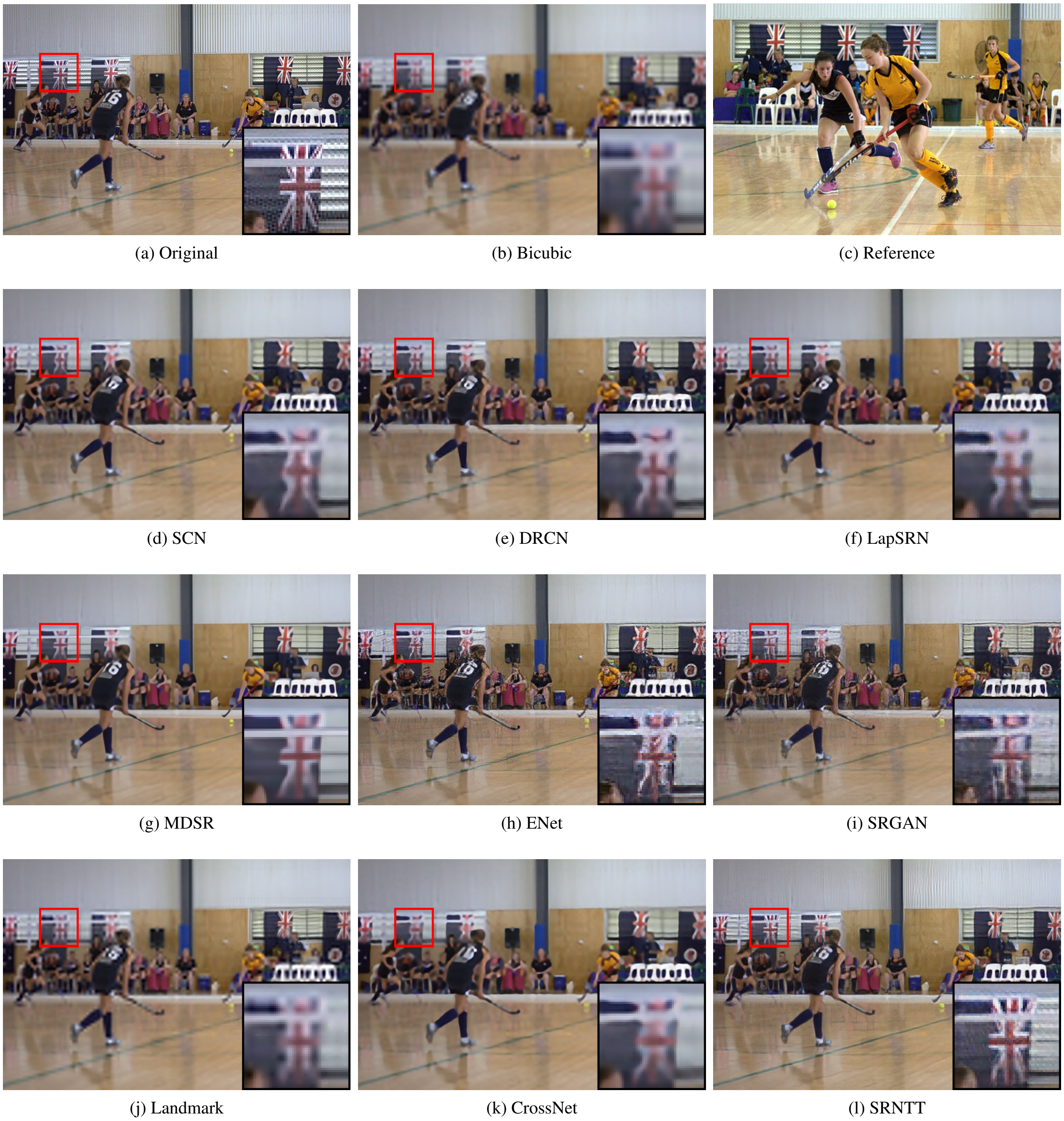
(i) SRGAN

(j) Landmark

(k) CrossNet

(l) SRNTT

Figure D.2: Comparison of different SR algorithms on CUFED5 dataset.



(a) Original

(b) Bicubic

(c) Reference

(d) SCN

(e) DRCN

(f) LapSRN

(g) MDSR

(h) ENet

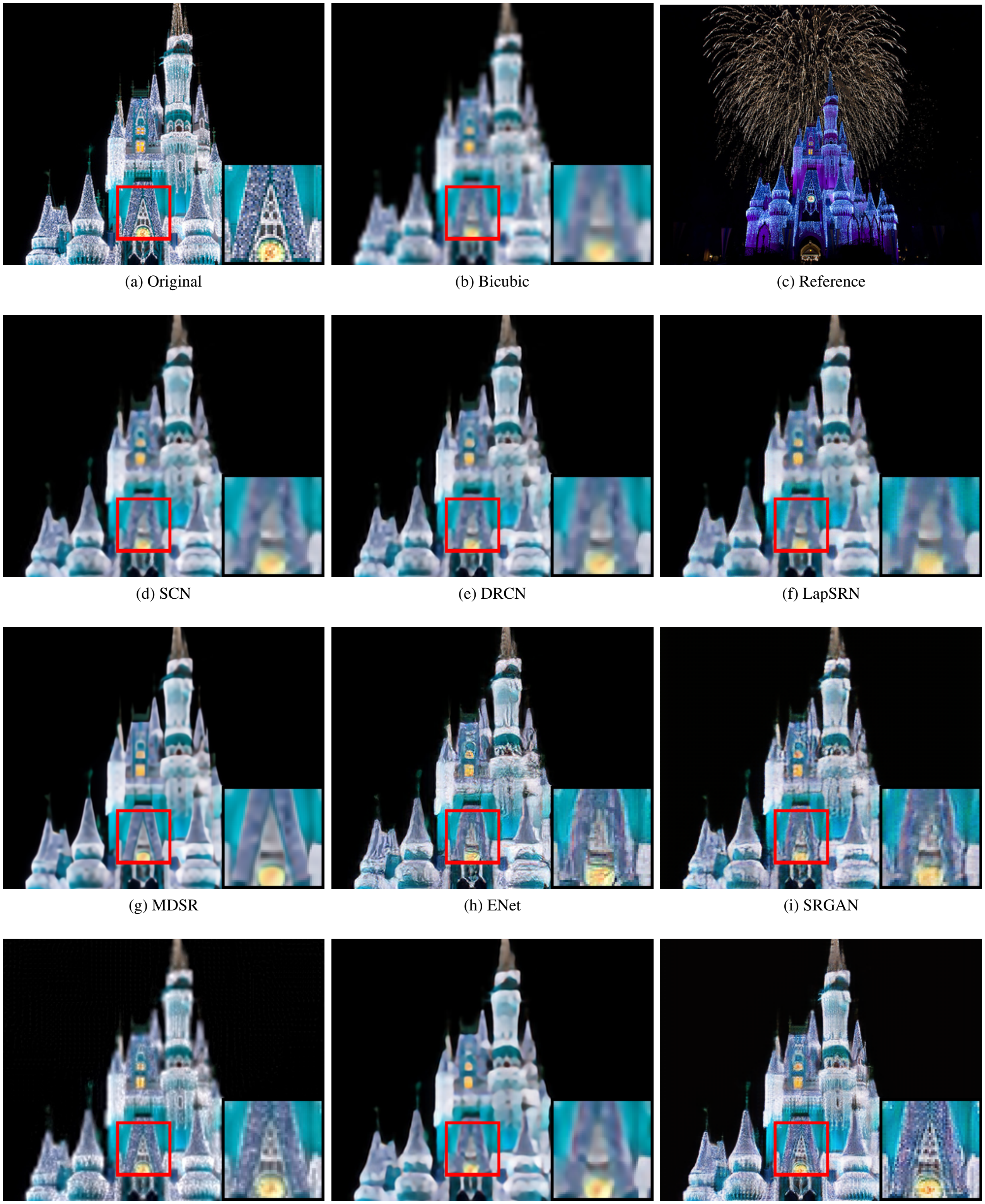
(i) SRGAN

(j) Landmark

(k) CrossNet

(l) SRNTT

Figure D.3: Comparison of different SR algorithms on CUFED5 dataset.



(a) Original

(b) Bicubic

(c) Reference

(d) SCN

(e) DRCN

(f) LapSRN

(g) MDSR

(h) ENet

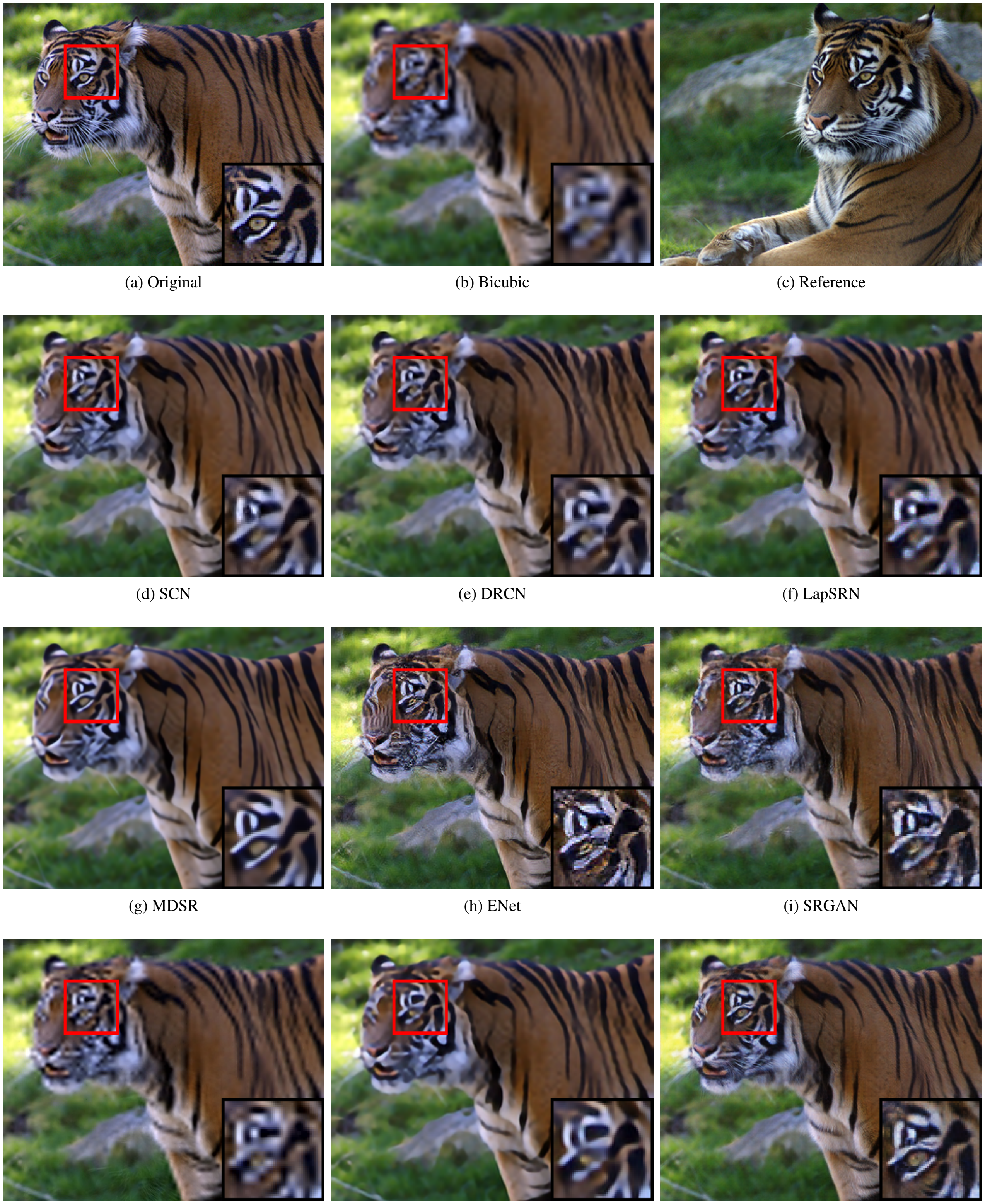
(i) SRGAN

(j) Landmark

(k) CrossNet

(l) SRNTT

Figure D.4: Comparison of different SR algorithms on CUFED5 dataset.



(a) Original

(b) Bicubic

(c) Reference

(d) SCN

(e) DRCN

(f) LapSRN

(g) MDSR

(h) ENet

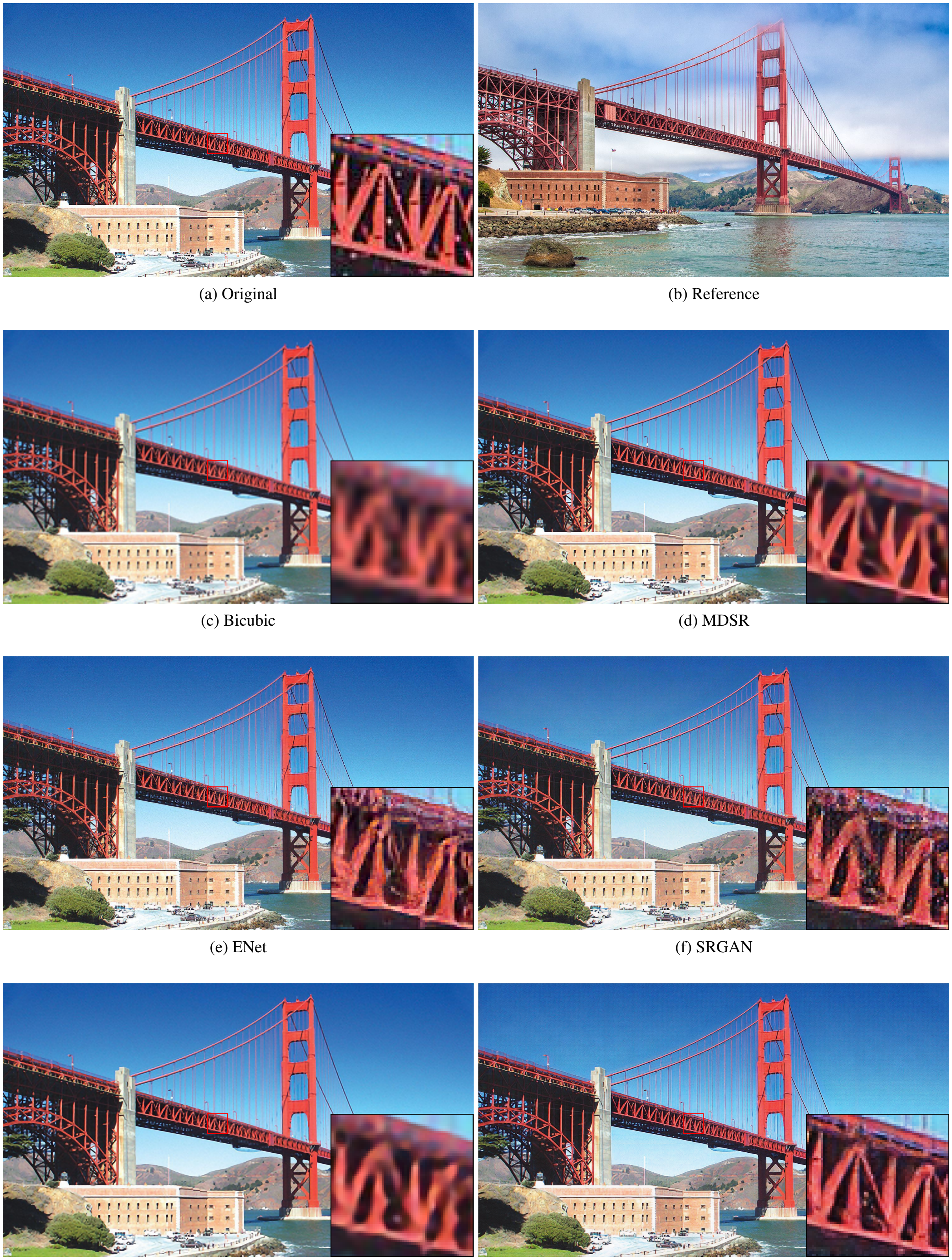
(i) SRGAN

(j) Landmark

(k) CrossNet

(l) SRNTT

Figure D.5: Comparison of different SR algorithms on CUFED5 dataset.



(a) Original

(b) Reference

(c) Bicubic

(d) MDSR

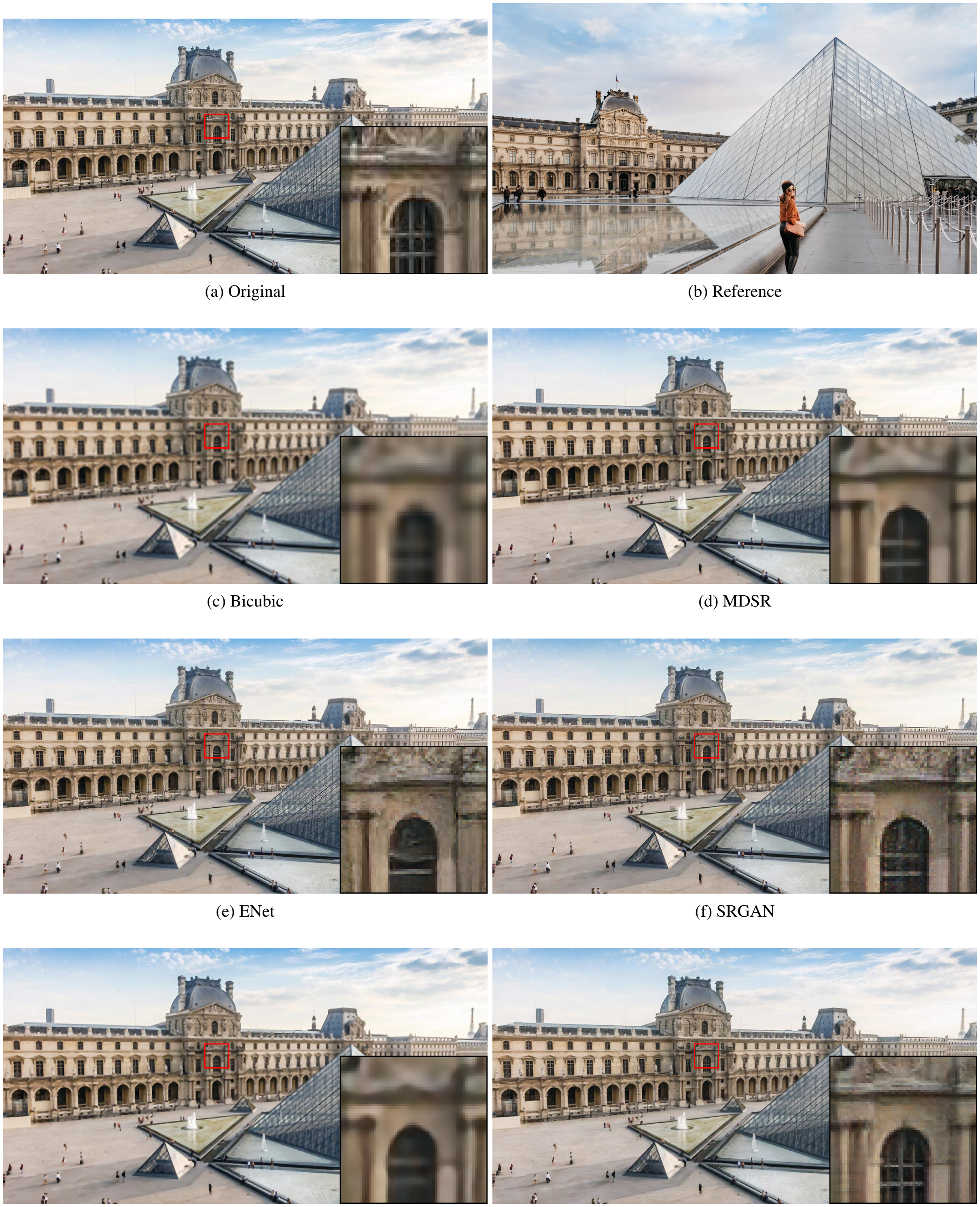
(e) ENet

(f) SRGAN

(g) CrossNet

(h) SRNTT

Figure E.1: Example of using the reference from web search.



(a) Original

(b) Reference

(c) Bicubic

(d) MDSR

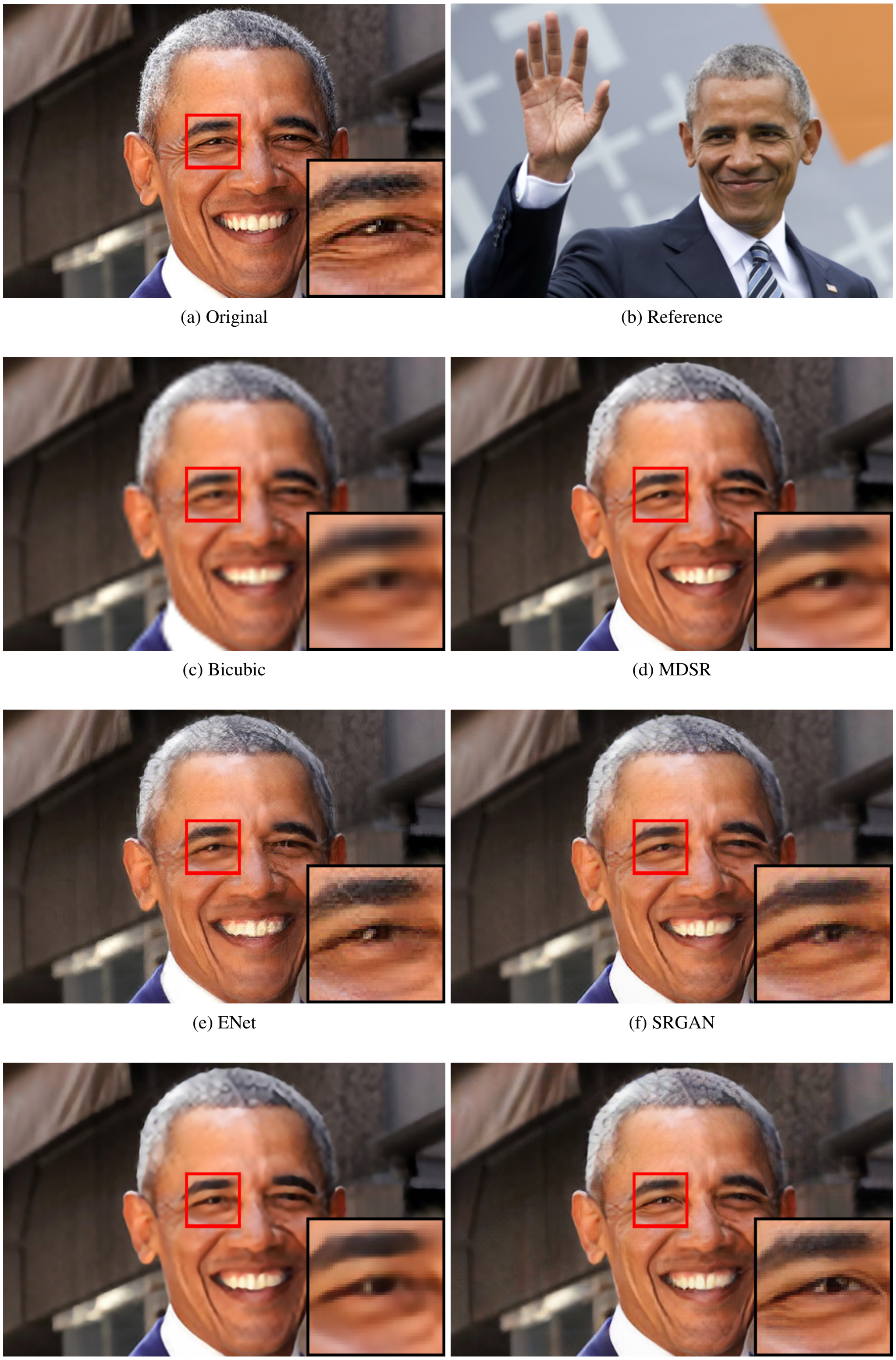
(e) ENet

(f) SRGAN

(g) CrossNet

(h) SRNTT

Figure E.2: Example of using the reference from web search.



(a) Original

(b) Reference

(c) Bicubic

(d) MDSR

(e) ENet

(f) SRGAN

(g) CrossNet

(h) SRNTT

Figure E.3: Example of using the reference from web search.