MASA-SR: Matching Acceleration and Spatial Adaptation for Reference-Based Image Super-Resolution

Liying Lu[1*] Wenbo Li[1*] Xin Tao[2] Jiangbo Lu[3] Jiaya Jia[1,3]
[1] The Chinese University of Hong Kong [2] Kuaishou [3] SmartMore
{lylu,wenboli,leojia}@cse.cuhk.edu.hk, jiangsutx@gmail.com, jiangbo@smartmore.com

Abstract

Reference-based image super-resolution (RefSR) has shown promising success in recovering high-frequency details by utilizing an external reference image (Ref). In this task, texture details are transferred from the Ref image to the low-resolution (LR) image according to their point- or patch-wise correspondence. Therefore, high-quality correspondence matching is critical. It is also desired to be computationally efficient. Besides, existing RefSR methods tend to ignore the potential large disparity in distributions between the LR and Ref images, which hurts the effectiveness of the information utilization. In this paper, we propose the MASA network for RefSR, where two novel modules are designed to address these problems. The proposed Match & Extraction Module significantly reduces the computational cost by a coarse-to-fine correspondence matching scheme. The Spatial Adaptation Module learns the difference of distribution between the LR and Ref images, and remaps the distribution of Ref features to that of LR features in a spatially adaptive way. This scheme makes the network robust to handle different reference images. Extensive quantitative and qualitative experiments validate the effectiveness of our proposed model.

1. Introduction

Single image super-resolution (SISR) is a fundamental computer vision task that aims to restore a high-resolution image (HR) with high-frequency details from its low-resolution counterpart (LR). Progress of SISR in recent years is based on deep convolutional neural networks (CNN) [3, 11, 12, 13, 16, 31]. Nevertheless, the ill-posed nature of SISR problems makes it still challenging to recover high-quality details.

In this paper, we explore reference-based super-resolution (RefSR), which utilizes an external reference image (Ref) to help super-resolve the LR image. Reference images usually contain similar content and texture with the LR image. They can be acquired from web image search or captured from different viewpoints. Transferring fine details to the LR image can overcome the limitation of SISR and has demonstrated promising performance in recent work of [35, 23, 34, 33, 30].

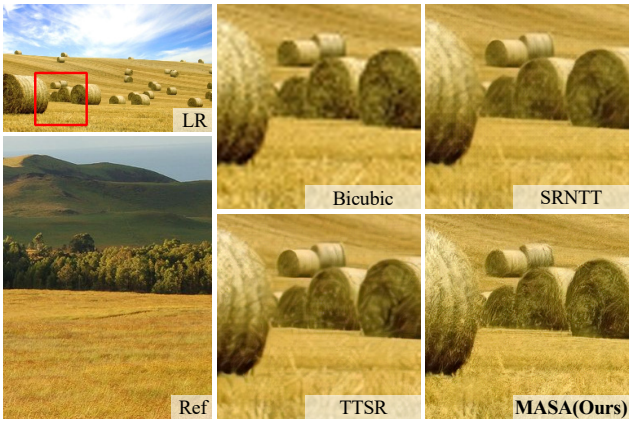


Figure 1: Visual comparison of $\times 4$ SR results. Our MASA method generates more appealing texture details than the two leading RefSR methods, *i.e.*, SRNTT [33] and TTSR [30].

Previous methods aimed at designing various ways to handle two critical issues in this task: *a) Correspond useful content in Ref images with LR images. b) Transfer features from Ref images to facilitate HR image reconstruction*. To address the first issue, methods perform spatial alignment between the Ref and LR images [35, 23] using optical flow or deformable convolutions [2, 36]. These alignment-based methods face challenges in, *e.g.*, finding long-distance correspondence. Other methods follow patch matching [34, 33, 30] in the feature space. State-of-the-art methods generally perform dense patch matching, leading to very high computational cost and large memory usage.

For the second issue, our finding is that even if LR and Ref images share similar content, color and luminance may

*Equal contribution

differ. Previous methods directly concatenate the LR features with the Ref ones and fuse them in convolution layers, which is not optimal.

To address the above problems, we propose a RefSR method called MASA-SR, which improves patch matching and transfer. The design of MASA has several advantages. First, the proposed Match & Extraction Module (MEM) performs correspondence matching in a coarse-to-fine manner, which largely reduces the computational cost while maintaining the matching quality. By leveraging the local coherence property of natural images, for each patch in the LR feature maps, we shrink its search space from the whole Ref feature map to a specific Ref block.

Second, the Spatial Adaptation Module is effective in handling the situations where there exists large disparity in color or luminance distribution between the LR and Ref images. It learns to remap the distribution of the Ref features to LR ones in a spatially adaptive way. Useful information in the Ref features thus can be transferred and utilized more effectively.

To the best of our knowledge, our model achieves state-of-the-art performance for the RefSR task. Our contributions are as follows.

- The proposed Match & Extraction Module significantly reduces the computational cost of correspondence matching in the deep feature space. Our results show that a *two-orders-of-magnitude* reduction measured in FLOPS is achieved.

- The proposed Spatial Adaptation Module is robust to Ref images with different color and luminance distributions. It enables the network to better utilize useful information extracted from Ref images.

2. Related Work

2.1. Single Image Super-Resolution

Effort has been made to improve performance of single image super resolution in recent years. In particular, deep learning based SISR methods achieved impressive success. Dong *et al*. [3] proposed the seminal CNN-based SISR model that consists of three convolution layers. Later, a variety of effective networks [11, 12, 13, 16, 31, 15, 26] were proposed for SR. With the help of residual learning, Kim *et al*. proposed VDSR [11] and DRCN [12] with deeper architectures and improved accuracy. Lai *et al*. [13] proposed LapSRN, which progressively reconstructs multi-scale results in a pyramid framework. Lim *et al*. [16] removed batch normalization layers in residual networks and further expanded the model size to improve SR performance. Zhang *et al*. [31] built a very deep network with residual in residual structure. Channel attention was introduced to model the inter-dependency across different channels.

Apart from MSE minimizing based methods, perception-driven ones received much attention. The perceptual loss [9] was introduced into SR tasks to enhance visual quality by minimizing errors on high-level features. Ledig *et al*. [14] proposed SRGAN, which was trained with an adversarial loss, generating photo-realistic images with natural details. To produce more perceptually satisfying results, ESRGAN [28] further improves SRGAN by introducing a relativistic adversarial loss. Different from the adversarial loss, the contextual loss [18, 19, 29] was proposed to maintain natural statistics in generated images by measuring the feature distribution.

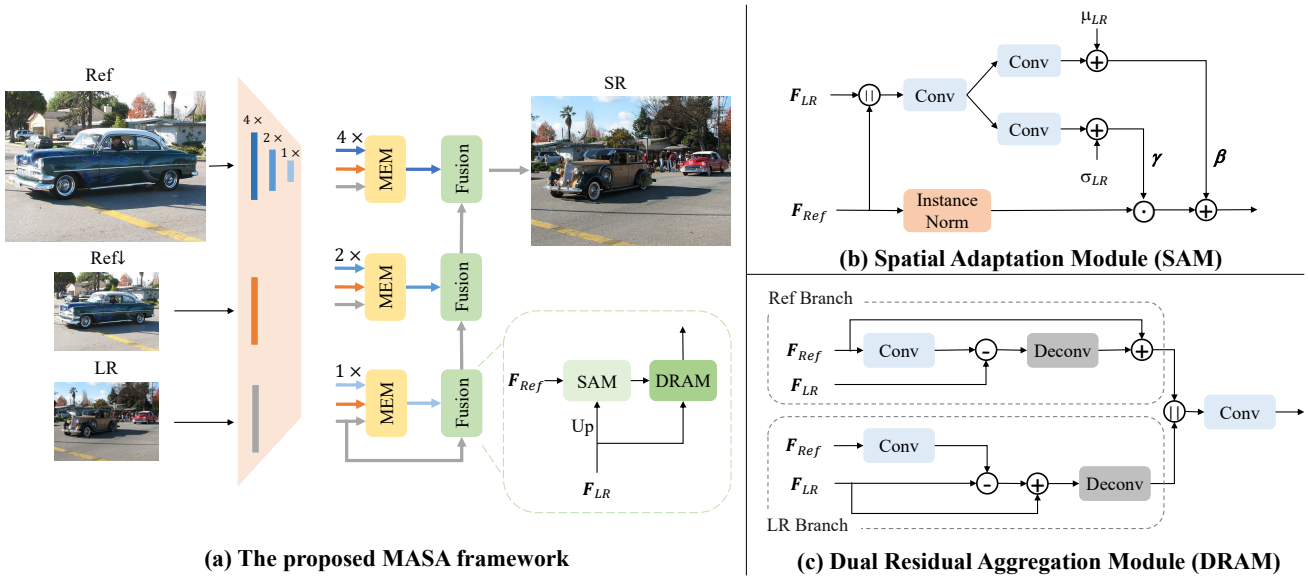
2.2. Reference-Based Super-Resolution

Compared with SISR, which only takes as input a low-resolution image, RefSR uses an additional reference image to upsample the LR input. The reference image generally has similar content with the LR image, capable to provide high-frequency details. Recent work mostly adopts CNN-based frameworks. One branch of RefSR performs spatial alignment between the Ref and LR images. CrossNet [35] estimated flow between the Ref and LR images at multi-scales and warped the Ref features according to the flow. However, the flow was obtained by a pre-trained network, leading to heavy computation and inaccurate estimation. Shim *et al*. [23] further proposed to align and extract Ref features by leveraging deformable convolutions [2, 36]. Nevertheless, these alignment-based methods are limited in finding long-distance correspondence.

Another branch follows the idea of patch matching [1]. Zheng *et al*. [34] trained two networks to learn feature correspondence and patch synthesis respectively. SRNTT [33] conducted multi-level patch matching between Ref and LR features extracted from the pre-trained VGG [24], and fused the swapped Ref features together with the LR features to generate the SR result. TTSR [30] further introduced the transformer architecture into the RefSR task and stacked the transformer in a cross-scale way to fuse multi-level information. The hard attention and soft attention in the transformer help transfer texture features from the Ref image more precisely. However, the patch matching method of SRNTT and TTSR is of high computation cost. They also leveraged VGG as the feature extractor that is heavy and requires pre-training.

3. MASA-SR Method

As shown in Fig. 2(a), our framework mainly consists of three parts: the encoder, *Matching & Extraction Modules (MEM)*, and fusion modules that contain *Spatial Adaptation Modules (SAM)* and *Dual Residual Aggregation Modules (DRAM)*. LR, Ref$\downarrow$ and Ref denote the low-resolution image, the $\times 4$ bicubic-downsampled reference image and the reference image, respectively. Unlike previ-



(a) The proposed MASA framework

(b) Spatial Adaptation Module (SAM)

(c) Dual Residual Aggregation Module (DRAM)

Figure 2: (a) Framework of the proposed MASA-SR, which consists of an encoder, several Match & Extraction Modules (MEM), Spatial Adaptation Modules (SAM) and Dual Residual Aggregation Modules (DRAM). (b) Structure of the Spatial Adaptation Modules (SAM), which is used to remap the distributions of the Ref features to that of the LR features. (c) Structure of the Dual Residual Aggregation Modules (DRAM), which is effective in feature fusion.

ous methods [33, 30] that use the pre-trained VGG as the feature extractor, our encoder is trained along with other parts of the network from scratch.

The encoder consists of three building blocks – the second and third blocks halve the size of the feature maps with stride 2. After passing the Ref image into the encoder, three Ref features with different scales are obtained as $\boldsymbol{F}^s_{Ref}$, where $s = 1, 2, 4$. The LR image and the Ref↓ image only go through the first block of the encoder, producing $\boldsymbol{F}_{LR}$ and $\boldsymbol{F}_{Ref\downarrow}$.

Afterwards, $\{\boldsymbol{F}_{LR}, \boldsymbol{F}_{Ref\downarrow}, \boldsymbol{F}^s_{Ref}\}$ are fed into MEM to perform coarse-to-fine correspondence matching and feature extraction as shown in Fig. 3. Though there are three MEMs in Fig. 2(a), the matching steps are only performed once between $\boldsymbol{F}_{LR}$ and $\boldsymbol{F}_{Ref\downarrow}$. The feature extraction stage is performed three times, each for one Ref feature $\boldsymbol{F}^s_{Ref}$ of scale s. To generate the final SR output, the LR features and output features from MEM are fused through the fusion module, where the proposed SAM is used to align the statistics of Ref features to those of LR ones. The proposed DRAM is used to enhance high-frequency details.

MEM, SAM and DRAM are explained in Sections 3.1, 3.2 and 3.3, respectively. The loss functions used to train the network are introduced in Section 3.4.

3.1. Matching & Extraction Module (MEM)

It is known that in a local region of a natural image, neighboring pixels are likely to come from common objects and share similar color statistics. Previous research on natural image priors also indicates that neighboring patches in one image are likely to find their correspondence spatially coherent with each other.

This motivates us to propose a coarse-to-fine matching scheme, *i.e.*, coarse block matching and fine patch matching. Note that 'block' and 'patch' are two different concepts in our method, and the size of block is larger than patch (3×3 in our experiments). As shown in Fig. 3, we first find correspondences in the feature space only for blocks. Specifically, we unfold the LR feature into non-overlapping blocks. Each LR block will find its most relevant Ref↓ block. By doing so, the computational cost of matching is reduced significantly compared with previous methods [33, 30]. To achieve enough precision, we further perform dense patch matching within each (LR block, Ref↓ block) pair. In the last stage, we extract useful Ref features according to the obtained correspondence information.

Stage 1: Coarse matching. In this stage, the LR feature $\boldsymbol{F}_{LR}$ is unfolded into K non-overlapping blocks: $\{\boldsymbol{B}^0_{LR}, ..., \boldsymbol{B}^{K-1}_{LR}\}$. For each LR block $\boldsymbol{B}^k_{LR}$, we find its most relevant Ref↓ block $\boldsymbol{B}^k_{Ref\downarrow}$.

We first take the center patch of $\boldsymbol{B}^k_{LR}$ to compute the cosine similarity with each patch of $\boldsymbol{F}_{Ref\downarrow}$ as

$$r^k_{c,j} = \left\langle \frac{\boldsymbol{p}^k_c}{\|\boldsymbol{p}^k_c\|}, \frac{\boldsymbol{q}_j}{\|\boldsymbol{q}_j\|} \right\rangle , \tag{1}$$

where $\boldsymbol{p}^k_c$ is the center patch of $\boldsymbol{B}^k_{LR}$, $\boldsymbol{q}_j$ is the j-th patch of $\boldsymbol{F}_{Ref\downarrow}$, and $r^k_{c,j}$ is their similarity score. According to

the similarity scores, we find the most similar patch for $\boldsymbol{p}_c^k$ in $\boldsymbol{F}_{Ref\downarrow}$. We then crop the block of size $d_x \times d_y$ centered around this similar patch, denoted as $\boldsymbol{B}_{Ref\downarrow}^k$. According to the local coherence property, for all patches in $\boldsymbol{B}_{LR}^k$, their most similar patches are likely to reside in this $\boldsymbol{B}_{Ref\downarrow}^k$. On the other hand, we also crop the corresponding $sd_x \times sd_y$ block from $\boldsymbol{F}_{Ref}^s$, denoted by $\boldsymbol{B}_{Ref}^{s,k}$, which will be used in the feature extraction stage.

Note that the center patch may not be representative enough to cover full content of the LR block if the size of the LR block is much larger than that of its center patch. This may mislead us to find the irrelevant Ref↓ block. To address it, we use center patches with different dilation rates to compute the similarity. The details are shown in Stage 1 of Fig. 3, where the dotted blue patch denotes the case of $dilation = 1$ and the dotted orange patch denotes the case of $dilation = 2$. Then the similarity score is computed as the sum of results of different dilations.

After this stage, for each LR block, we obtain its most relevant Ref↓ block and the corresponding Ref block, forming triples of ($\boldsymbol{B}_{LR}^k$, $\boldsymbol{B}_{Ref\downarrow}^k$, $\boldsymbol{B}_{Ref}^{s,k}$). We limit the search space of $\boldsymbol{B}_{LR}^k$ to $\boldsymbol{B}_{Ref\downarrow}^k$ in the fine matching stage.

Stage 2: Fine matching. In this stage, dense patch matching is performed between each LR block and its corresponding Ref↓ block independently. A set of index maps $\{\boldsymbol{D}^0, ..., \boldsymbol{D}^{K-1}\}$ and similarity maps $\{\boldsymbol{R}^0, ..., \boldsymbol{R}^{K-1}\}$ are obtained.

More precisely, taking the k-th pair ($\boldsymbol{B}_{LR}^k$, $\boldsymbol{B}_{Ref\downarrow}^k$) for example, we compute the similarity score between each patch of $\boldsymbol{B}_{LR}^k$ and each patch of $\boldsymbol{B}_{Ref\downarrow}^k$ as

$$r_{i,j}^k = \left\langle \frac{\boldsymbol{p}_i^k}{\|\boldsymbol{p}_i^k\|}, \frac{\boldsymbol{q}_j^k}{\|\boldsymbol{q}_j^k\|} \right\rangle, \tag{2}$$

where $\boldsymbol{p}_i^k$ is the i-th patch of $\boldsymbol{B}_{LR}^k$, $\boldsymbol{q}_j^k$ is the j-th patch of $\boldsymbol{B}_{Ref\downarrow}^k$, and $r_{i,j}^k$ is their similarity score. Then the i-th element of $\boldsymbol{D}^k$ is calculated as

$$\boldsymbol{D}_i^k = \arg\max_j \ r_{i,j}^k \ . \tag{3}$$

The i-th element of $\boldsymbol{R}^k$ is the highest similarity score related to the i-th patch of $\boldsymbol{B}_{LR}^k$ as

$$\boldsymbol{R}_i^k = \max_j \ r_{i,j}^k \ . \tag{4}$$

Stage 3: Feature extraction. In this stage, we first extract patches from $\boldsymbol{B}_{Ref}^{s,k}$ according to the index map $\boldsymbol{D}^k$, and form a new feature map $\boldsymbol{B}_M^{s,k}$. Specifically, We crop the $\boldsymbol{D}_i^k$-th patch of $\boldsymbol{B}_{Ref}^{s,k}$ as the i-th patch of $\boldsymbol{B}_M^{s,k}$. Moreover, since Ref features with higher similarity scores are more useful, we multiply $\boldsymbol{B}_M^{s,k}$ with the corresponding similarity score map $\boldsymbol{R}^k$ to get the weighted feature block as

$$\boldsymbol{B}_M^{s,k} := \boldsymbol{B}_M^{s,k} \odot (\boldsymbol{R}^k) \uparrow , \tag{5}$$

where $()\uparrow$ and $\odot$ denote bilinear interpolation and element-wise multiplication.

The final ouput of MEM is obtained by folding $\{\boldsymbol{B}_M^{s,0}, ..., \boldsymbol{B}_M^{s,K-1}\}$ together, which is the reverse operation of the unfolding operation in Stage 1.

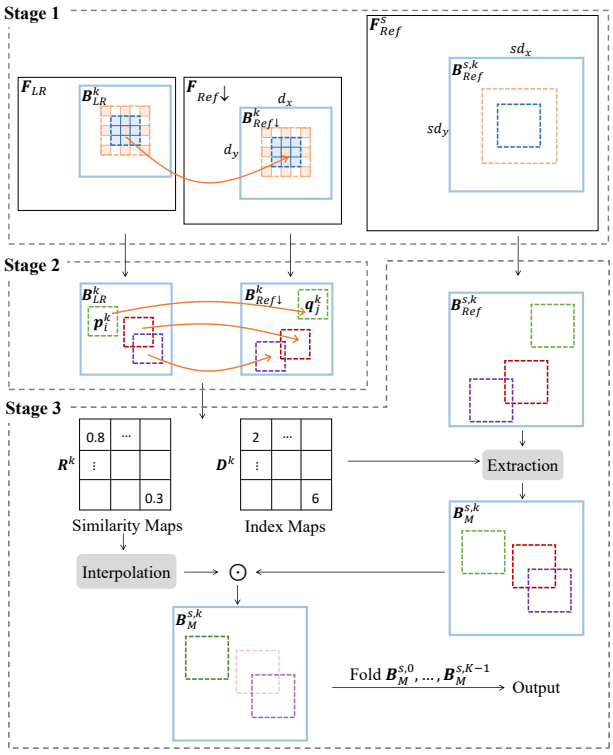


Figure 3: Pipeline of the Match & Extraction Module (MEM). In **Stage 1**, each LR block finds the most relevant Ref↓ block. In **Stage 2**, dense patch matching is performed in each (LR block, Ref↓ block) pair. In **Stage 3**, Ref features are extracted according to the similarity and index maps produced in the second stage. All the blocks are denoted by solid squares in light blue and patches are denoted by dotted squares in different colors.

Analysis. For an LR image with m pixels and a Ref↓ image with n pixels, computational complexity of matching in previous methods is generally $\mathrm{O}(mn)$. While in the MEM, suppose each Ref↓ block has n' pixels, computational complexity is reduced to $\mathrm{O}(Kn + mn')$. Since K is much smaller than m and n' is also several hundred times smaller than n, the computational cost is reduced significantly through this coarse-to-fine matching scheme.

3.2. Spatial Adaptation Module (SAM)

In many situations, the LR and the Ref images may have similar content and texture. But color and luminance dis-

tributions diverge. Thus the distribution of extracted Ref features may not be consistent with that of the LR features. Therefore, simply concatenating the Ref and LR features together and feeding them into the following convolution layers is not optimal. Inspired by [6, 21], we propose the Spatial Adaptation Module (SAM) to remap the distribution of the extracted Ref features to that of the LR features.

We illustrate the structure of SAM in Fig. 2b. The LR feature and extracted Ref feature are first concatenated before feeding into convolution layers to produce two parameters $\boldsymbol{\beta}$ and $\boldsymbol{\gamma}$, which are with the same size as the LR feature. Then instance normalization [27] is applied to the Ref feature as

$$\boldsymbol{F}^c_{Ref} \longleftarrow \frac{\boldsymbol{F}^c_{Ref} - \boldsymbol{\mu}^c_{Ref}}{\boldsymbol{\sigma}^c_{Ref}} , \tag{6}$$

where $\boldsymbol{\mu}^c_{Ref}$ and $\boldsymbol{\sigma}^c_{Ref}$ are the mean and standard deviation of $\boldsymbol{F}_{Ref}$ in channel c as

$$\boldsymbol{\mu}^c_{Ref} = \frac{1}{HW}\sum_{y,x} \boldsymbol{F}^{c,y,x}_{Ref} , \tag{7}$$

$$\boldsymbol{\sigma}^c_{Ref} = \sqrt{\frac{1}{HW}\sum_{y,x} \left(\boldsymbol{F}^{c,y,x}_{Ref} - \boldsymbol{\mu}^c_{Ref}\right)^2} . \tag{8}$$

H and W are the height and width of $\boldsymbol{F}_{Ref}$ .We then update $\boldsymbol{\beta}$ and $\boldsymbol{\gamma}$ with the mean and standard deviation of the LR feature of

$$\boldsymbol{\beta} \longleftarrow \boldsymbol{\beta} + \boldsymbol{\mu}_{LR} , \tag{9}$$

$$\boldsymbol{\gamma} \longleftarrow \boldsymbol{\gamma} + \boldsymbol{\sigma}_{LR} , \tag{10}$$

where $\boldsymbol{\mu}_{LR}$ and $\boldsymbol{\sigma}_{LR}$ are computed in a similar way as Eqs. (7) and (8). Finally, $\boldsymbol{\gamma}$ and $\boldsymbol{\beta}$ are multiplied and added to the normalized Ref feature in an element-wise manner as

$$\boldsymbol{F}_{Ref} \longleftarrow \boldsymbol{F}_{Ref} \cdot \boldsymbol{\gamma} + \boldsymbol{\beta} . \tag{11}$$

Since the difference between the Ref features and LR features varies with respect to the spatial location, while the statistics $\boldsymbol{\mu}_{LR}$, $\boldsymbol{\sigma}_{LR}$, $\boldsymbol{\mu}_{Ref}$ and $\boldsymbol{\sigma}_{Ref}$ are of size $C \times 1 \times 1$, we use learnable convolutions to predict two spatial-wise adaptation parameters $\boldsymbol{\beta}$ and $\boldsymbol{\gamma}$. Unlike [21] that only uses the segmentation maps to produce two parameters, the convolutions in SAM takes as the input both Ref and LR features to learn their difference. Besides, after obtaining $\boldsymbol{\beta}$ and $\boldsymbol{\gamma}$ from the convolutions, we add them with the mean and standard deviation of the LR features.

3.3. Dual Residual Aggregation Module (DRAM)

After spatial adaptation, the transferred Ref features are fused with the LR features using our proposed Dual Residual Aggregation Module (DRAM) as shown in Fig. 2(c). DRAM consists of two branches, *i.e.*, the LR branch and the Ref branch.

The Ref branch aims to refine the high-frequency details of the Ref features. It first downsamples the Ref feature $\boldsymbol{F}_{Ref}$ by a convolution layer with stride 2, and the residual $\boldsymbol{Res}_{Ref}$ between the downsampled Ref feature and the LR feature $\boldsymbol{F}_{LR}$ is then upsampled by a transposed convolution layer as

$$\begin{cases} \boldsymbol{Res}_{Ref} = Conv(\boldsymbol{F}_{Ref}) - \boldsymbol{F}_{LR} , \\ \boldsymbol{F}'_{Ref} = \boldsymbol{F}_{Ref} + Deconv(\boldsymbol{Res}_{Ref}) . \end{cases} \tag{12}$$

Similarly, the high-frequency details of the LR features are refined as

$$\begin{cases} \boldsymbol{Res}_{LR} = \boldsymbol{F}_{LR} - Conv(\boldsymbol{F}_{Ref}) , \\ \boldsymbol{F}'_{LR} = Deconv(\boldsymbol{F}_{LR} + \boldsymbol{Res}_{LR}) . \end{cases} \tag{13}$$

At last, the outputs of two branches are concatenated and passed through another convolution layer with stride 1. In this way, the details in the LR and Ref features are enhanced and aggregated, leading to more representative features.

3.4. Loss Functions

Reconstruction loss. We adopt L_1 loss as the reconstruction loss as

$$\mathcal{L}_{rec} = \|\boldsymbol{I}_{HR} - \boldsymbol{I}_{SR}\|_1 , \tag{14}$$

where $\boldsymbol{I}_{HR}$ and $\boldsymbol{I}_{SR}$ denote the ground truth image and the network output.

Perceptual loss. The perceptual loss is expressed as

$$\mathcal{L}_{per} = \|\phi_i(\boldsymbol{I}_{HR}) - \phi_i(\boldsymbol{I}_{SR})\|_2 , \tag{15}$$

where ϕ_i denotes the i-th layer of VGG19. Here we use $conv5\_4$.

Adversarial loss. The adversarial loss [4] $\mathcal{L}_{adv}$ is effective in generating visually pleasing images with natural details. We adopt the Relativistic GANs [10]:

$$\begin{aligned} \mathcal{L}_D = &- \mathbb{E}_{\boldsymbol{I}_{HR}}\left[\log(D(\boldsymbol{I}_{HR}, \boldsymbol{I}_{SR}))\right] \\ &- \mathbb{E}_{\boldsymbol{I}_{SR}}\left[\log(1 - D(\boldsymbol{I}_{SR}, \boldsymbol{I}_{HR}))\right] , \end{aligned} \tag{16}$$

$$\begin{aligned} \mathcal{L}_G = &- \mathbb{E}_{\boldsymbol{I}_{HR}}\left[\log(1 - D(\boldsymbol{I}_{HR}, \boldsymbol{I}_{SR}))\right] \\ &- \mathbb{E}_{\boldsymbol{I}_{SR}}\left[\log(D(\boldsymbol{I}_{SR}, \boldsymbol{I}_{HR}))\right] . \end{aligned} \tag{17}$$

Full objective. Our full objective is defined as

$$\mathcal{L} = \lambda_{rec}\mathcal{L}_{rec} + \lambda_{per}\mathcal{L}_{per} + \lambda_{adv}\mathcal{L}_{adv} . \tag{18}$$

4. Experiments

4.1. Datasets

Our model is trained on CUFED5 [33] dataset with a $\times 4$ upscale factor following the setting in [33, 30]. CUFED5

Algorithm	CUFED5	Sun80	Urban100
SRCNN [3]	25.33 / 0.745	28.26 / 0.781	24.41 / 0.738
MDSR [16]	25.93 / 0.777	28.52 / 0.792	25.51 / 0.783
RDN [32]	25.95 / 0.769	29.63 / 0.806	25.38 / 0.768
RCAN [31]	26.15 / 0.767	29.86 / 0.808	25.40 / 0.765
HAN [20]	26.15 / 0.767	29.91 / 0.809	25.41 / 0.765
SRGAN [14]	24.40 / 0.702	26.76 / 0.725	24.07 / 0.729
ENet [22]	24.24 / 0.695	26.24 / 0.702	23.63 / 0.711
ESRGAN [28]	23.84 / 0.693	26.77 / 0.705	23.25 / 0.695
CrossNet [35]	25.48 / 0.764	28.52 / 0.793	25.11 / 0.764
SRNTT [33]	25.61 / 0.764	27.59 / 0.756	25.09 / 0.774
SRNTT-rec [33]	26.24 / 0.784	28.54 / 0.793	25.50 / 0.783
TTSR [30]	25.53 / 0.765	28.59 / 0.774	24.62 / 0.747
TTSR-rec [30]	27.09 / 0.804	30.02 / 0.814	25.87 / 0.784
MASA	24.92 / 0.729	27.12 / 0.708	23.78 / 0.712
MASA-rec	27.54 / 0.814	30.15 / 0.815	26.09 / 0.786

Table 1: PSNR/SSIM comparison among different SR methods on 3 testing datasets. Methods are grouped by SISR (top) and RefSR (bottom). The best and the second best results are colored in red and blue.

is composed of 11,871 training pairs. Each pair contains an original HR image and a corresponding reference image at 160 $\times$ 160 resolution. To validate the generalization capacity of our model, we test it on three popular benchmarks: CUFED5 testing set, Urban100 [7] and Sun80 [25].

CUFED5 testing set consists of 126 testing pairs, and each HR image is accompanied by 4 reference images with different similarity levels based on SIFT [17] feature matching. We stitch 4 references to one image, same as that of [30], during testing. Urban100 contains 100 building images without references, and we take the LR image as the reference such that the network explores self-similarity of input images. Sun80 contains 80 natural images, each paired with several references. We randomly sample one of them as the reference image. All results of PSNR and SSIM are evaluated on the Y channel of YCbCr color space.

4.2. Implementation Details

The encoder consists of 3 building blocks, each composed of 1 convolutional layer and 4 ResBlocks [5]. The fusion module consists of 1 spatial adaptation module, 1 dual residual aggregation module, several convolutional layers and ResBlocks. The numbers of ResBlocks in $1\times$, $2\times$ and $4\times$ fusion modules are 12, 8, and 4. The number of all intermediate channels is 64. The activation function is ReLU. No batch normalization (BN) layer is used in our network.

In MEM, the LR block size is set to 8×8. The Ref$\downarrow$ block size is set to $\frac{12H_{Ref\downarrow}}{H_{LR}} \times \frac{12W_{Ref\downarrow}}{W_{LR}}$, where H_{LR}, W_{LR} and $H_{Ref\downarrow}$, $W_{Ref\downarrow}$ are the height and width of the LR image and the Ref$\downarrow$ image, respectively. The patch size is set to 3×3. The discriminator structure is the same as that adopted in [28]. We train our model with the Adam optimizer by setting $\beta_1 = 0.9$ and $\beta_2 = 0.999$. The learning rate is set to 1e-4 and the batch size is 9. The weight coefficients λ_{rec}, λ_{per} and λ_{adv} are 1, 1 and 5e-3, respectively.

Algorithm	FLOPS-M (G)	FLOPS-T (G)	Param. (M)	Runtime (ms)
CrossNet [35]	-	348.31	35.18	98.7
SRNTT [33]	6,005.78	6,500.70	5.75	4,161.6
TTSR [30]	618.48	1,044.28	6.99	199.8
MASA	8.84	367.93	4.03	141.1

Table 2: FLOPS of matching steps (FLOPS-M), total FLOPS (FLOPS-T), number of network parameters and runtime comparisons among different RefSR methods. CrossNet [35] is an alignment-based method, while the others are matching-based methods.

4.3. Comparison with State-of-the-Art Methods

We compare our proposed model with previous state-of-the-art SISR and RefSR methods. SISR methods include SRCNN [3], MDSR [16], RDN [32], RCAN [31] and HAN [20]. GAN-based SISR methods include SRGAN [14], ENet [22] and ESRGAN [28]. Among all these methods, RCAN and HAN achieved the best performance on PSNR, and ESRGAN is considered state-of-the-art in terms of visual quality. Some recent RefSR methods are also included, *i.e.*, CrossNet [35], SRNTT [33], TTSR [30]. All the models are trained on the CUFED5 training set, and tested on the CUFED5 testing set of Sun80 and Urban100. The scale factor in all experiments is $\times4$.

Quantitative evaluations. For fair comparison with other MSE minimization based methods on PSNR and SSIM, we train another version of MASA by only minimizing the reconstruction loss, denoted as MASA-rec.

Table 1 shows the quantitative comparisons on PSNR and SSIM, where the best and the second best results are colored in red and blue. As shown in Table 1, our model outperforms state-of-the-art methods on all three testing sets.

We also compare the FLOPS, the number of network parameters and runtime with other RefSR methods in Table 2, where FLOPS-M denotes the FLOPS of the matching steps, and FLOPS-T denotes the total FLOPS. The FLOPS is calculated on input of a 128 $\times$ 128 LR image and a 512 $\times$ 512 Ref image. Our model yields the smallest number of parameters and the second-best FLOPS and runtime with the best performance on PSNR/SSIM. Though the alignment-based method CrossNet [35] has the smallest FLOPS and runtime, its performance on PSNR/SSIM is not on top when compared with other methods in Table 1.

Qualitative evaluations. We show visual comparison between our model and other SISR and RefSR methods in Fig. 4. Our proposed MASA outperforms other methods in terms of visual quality, generating more fine details without introducing many unpleasing artifacts in general. MASA produces a higher level of natural hair, wrinkle, and leaf texture.

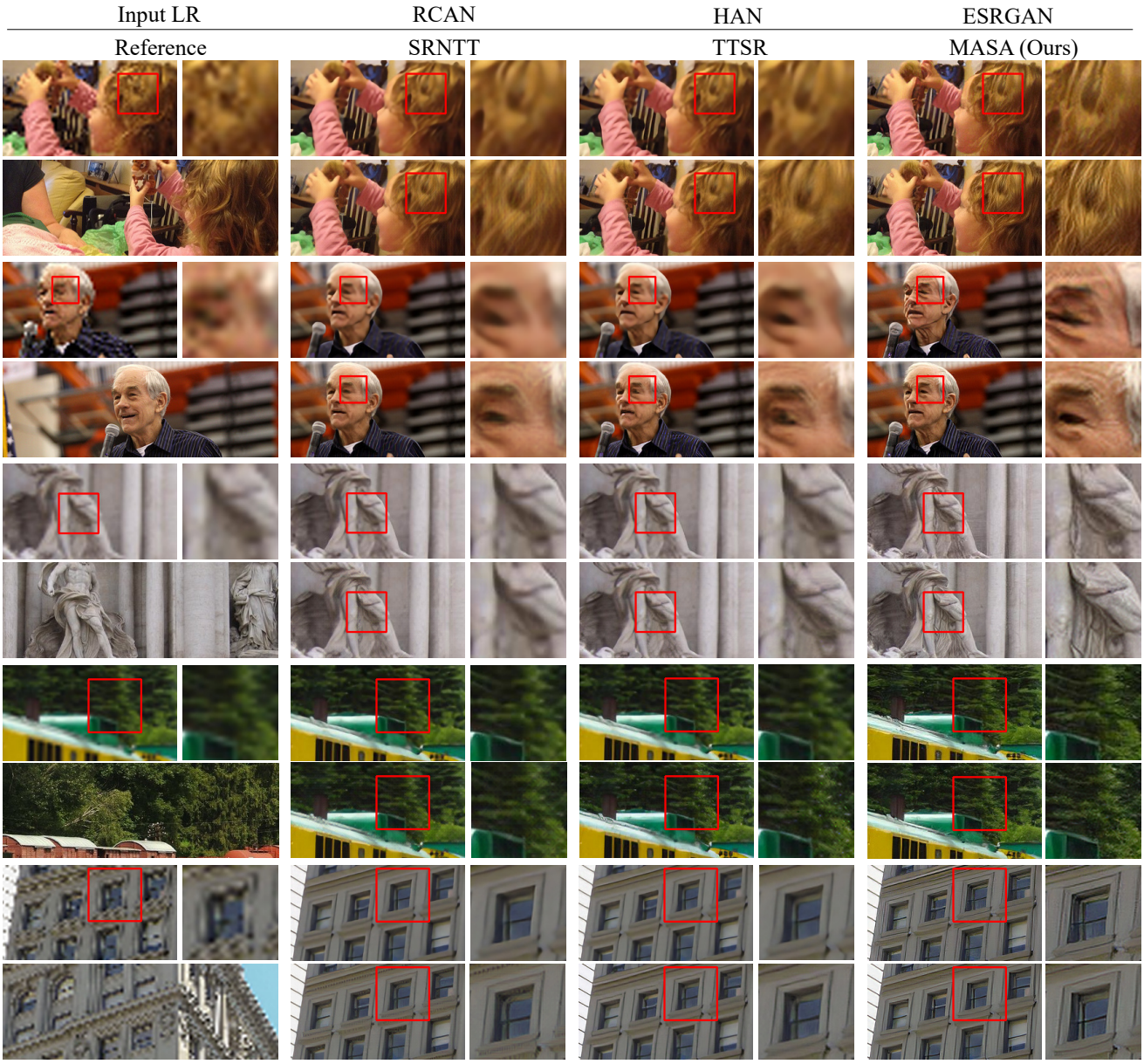


Figure 4: Visual comparison among different SR methods on the CUFED5 testing set (top two examples), Sun80 [25] (the third and the fourth example) and Urban100 [7] (the last example). This figure is best viewed by zoom-in.

4.4. Ablation Study

In this section, we conduct several ablation studies to investigate our proposed method. We analyze the influence of different block sizes and dilation rates used in the coarse matching stage. We also verify the effectiveness of the proposed Spatial Adaptation Module and the Dual Residual Aggregation Module.

Influence of block sizes and dilation rates. In the matching & extraction module, the LR block size, the Ref↓ block size and the dilation rates are key factors to balance the matching accuracy and efficiency. Thus we analyze the influence of these three hyper-parameters on the CUFED5 testing set. Fig. 5 shows the ablation results. We only show the FLOPS of batch matrix-matrix product operations.

Fig. 5(a) shows the influence of the LR block size. It can be seen that as the LR block size increases, PSNR and FLOPS both drop, indicating the decreasing matching accuracy and computational cost. We also test the case that the size of LR block is 1×1, the PSNR reaches 27.60 dB while the FLOPS sharply goes up to 787.77G, which is not shown in Fig. 5(a).

Fig. 5(b) shows the influence of the Ref↓ block size.

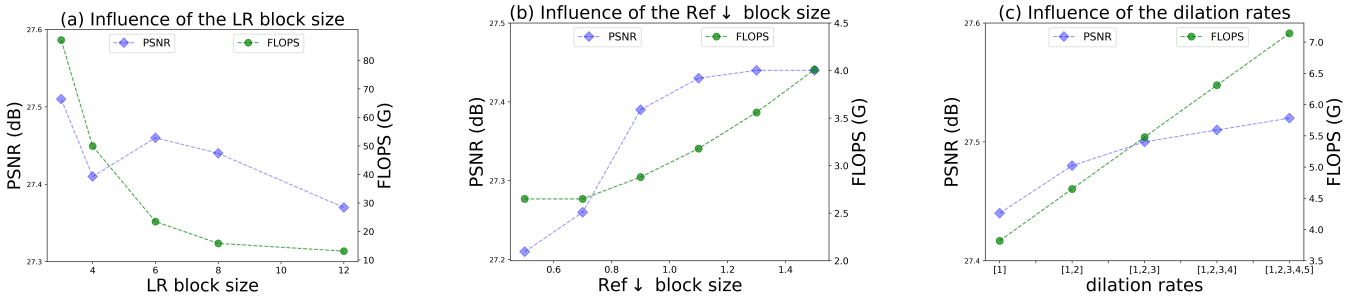


Figure 5: Influence of different LR block sizes, Ref↓ block sizes and dilation rates on PSNR and FLOPS. (a) Influence of LR block sizes. (b) Influence of Ref↓ block sizes. (c) Influence of dilation rates.

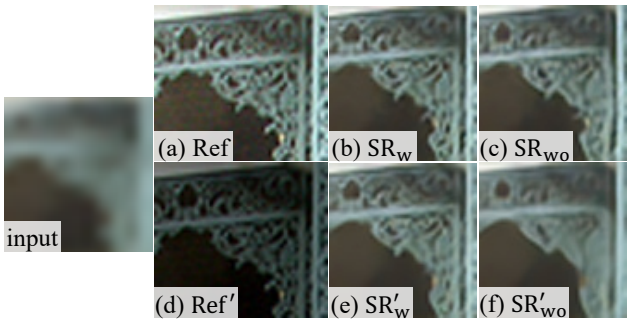


Figure 6: Ablation study on the spatial adaptation module. The results with the subscript 'w' are the outputs of the model with SAM, while those with the subscript 'wo' are the outputs of the baseline model without SAM.

The PSNR and FLOPS increase as the increasing of the Ref↓ block size. Because larger Ref↓ block size boosts the matching accuracy in the fine matching stage. However, when the Ref↓ block size increases to some extent, the growth of PSNR slows down. On the other hand, since patch matching has to be performed on larger blocks, the computational cost increases inevitably.

As illustrated in Fig. 5(c), the more combinations of different dilation rates exist, the higher PSNR can be obtained. Since larger dilation rates cover a larger area in the LR block, it leads to more accurate coarse matching.

Effect of spatial adaptation module. The spatial adaptation module plays the role of aligning the distribution of the Ref features to that of the LR features. As shown in Table 3, compared with the baseline (without any normalization), SAM improves the PSNR and SSIM by 0.22 dB and 0.007, *only* introducing 0.15M more parameters. We also compare SAM with other normalization methods, including adaptive instance norm. (AdaIN) [8] and SPADE [21]. We found that SPADE is not suitable for this task, and the performance of AdaIN is similar with that of the baseline.

Qualitative comparison is visualized in Fig. 6. We first test the model with SAM and the baseline model on the Ref image, which has a similar luminance with the LR image.

	baseline	AdaIN	SPADE	SAM (ours)
PSNR / SSIM	27.32 / 0.807	27.30 / 0.806	24.46 / 0.688	27.54 / 0.814
param.	3.88M	3.88M	3.99M	4.03M

Table 3: Ablation study on the spatial adaptation module.

Model	Model 1	Model 2	Model 3	Model 4
Ref branch	✗	✗	✓	✓
LR branch	✗	✓	✗	✓
PSNR / SSIM	27.43 / 0.809	27.51 / 0.813	27.48 / 0.811	27.54 / 0.814
param.	3.73M	3.88M	3.88M	4.03M

Table 4: Ablation study on dual residual aggregation.

Then we change the luminance of the Ref image (denoted by Ref′), and test two models on it. As shown in Fig. 6, the model with SAM performs better in both cases. Besides, when testing on Ref′ with changing luminance, the performance of the model with SAM almost keeps the same, while the performance of baseline drops significantly. This demonstrates that SAM is robust in handling different Ref images.

Effect of dual residual aggregation module. To verify the effectiveness of the dual residual aggregation module (DRAM), we conduct ablation study on 4 models as shown in Table 4. Model 1 simply concatenates the LR features with the Ref features and feeds them into a convolution layer. Model 2 only keeps the LR branch of the DRAM, and Model 3 only keeps the Ref branch. Model 4 is the proposed DRAM. In Table 4, it is clear that DRAM outperforms Model 1 by 0.11 dB.

5. Conclusion

In this paper, we have proposed MASA-SR, a new end-to-end trainable network for RefSR. It features a coarse-to-fine correspondence matching scheme to reduce the computational cost significantly, while achieving strong matching and transfer capability. Further, a novel Spatial Adaptation Module is designed to boost the robustness of the network when dealing with Ref images with different distributions. Our method achieves state-of-the-art results both quantitatively and qualitatively across different datasets.

References

[1] Connelly Barnes, Eli Shechtman, Adam Finkelstein, and Dan B Goldman. Patchmatch: A randomized correspondence algorithm for structural image editing. *ACM Trans. Graph.*, 28(3):24, 2009.

[2] Jifeng Dai, Haozhi Qi, Yuwen Xiong, Yi Li, Guodong Zhang, Han Hu, and Yichen Wei. Deformable convolutional networks. In *ICCV*, pages 764–773, 2017.

[3] Chao Dong, Chen Change Loy, Kaiming He, and Xiaoou Tang. Image super-resolution using deep convolutional networks. *TPAMI*, 38(2):295–307, 2015.

[4] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. In *NeurIPS*, pages 2672–2680, 2014.

[5] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *CVPR*, pages 770–778, 2016.

[6] Aaron Hertzmann, Charles E Jacobs, Nuria Oliver, Brian Curless, and David H Salesin. Image analogies. In *Proceedings of the 28th annual conference on Computer graphics and interactive techniques*, pages 327–340, 2001.

[7] Jia-Bin Huang, Abhishek Singh, and Narendra Ahuja. Single image super-resolution from transformed self-exemplars. In *CVPR*, pages 5197–5206, 2015.

[8] Xun Huang and Serge Belongie. Arbitrary style transfer in real-time with adaptive instance normalization. In *ICCV*, pages 1501–1510, 2017.

[9] Justin Johnson, Alexandre Alahi, and Li Fei-Fei. Perceptual losses for real-time style transfer and super-resolution. In *ECCV*, pages 694–711. Springer, 2016.

[10] Alexia Jolicoeur-Martineau. The relativistic discriminator: a key element missing from standard gan. *arXiv preprint arXiv:1807.00734*, 2018.

[11] Jiwon Kim, Jung Kwon Lee, and Kyoung Mu Lee. Accurate image super-resolution using very deep convolutional networks. In *CVPR*, pages 1646–1654, 2016.

[12] Jiwon Kim, Jung Kwon Lee, and Kyoung Mu Lee. Deeply-recursive convolutional network for image super-resolution. In *CVPR*, pages 1637–1645, 2016.

[13] Wei-Sheng Lai, Jia-Bin Huang, Narendra Ahuja, and Ming-Hsuan Yang. Deep laplacian pyramid networks for fast and accurate super-resolution. In *CVPR*, pages 624–632, 2017.

[14] Christian Ledig, Lucas Theis, Ferenc Huszár, Jose Caballero, Andrew Cunningham, Alejandro Acosta, Andrew Aitken, Alykhan Tejani, Johannes Totz, Zehan Wang, et al. Photo-realistic single image super-resolution using a generative adversarial network. In *CVPR*, pages 4681–4690, 2017.

[15] Wenbo Li, Kun Zhou, Lu Qi, Nianjuan Jiang, Jiangbo Lu, and Jiaya Jia. Lapar: Linearly-assembled pixel-adaptive regression network for single image super-resolution and beyond. *NeurIPS*, 33, 2020.

[16] Bee Lim, Sanghyun Son, Heewon Kim, Seungjun Nah, and Kyoung Mu Lee. Enhanced deep residual networks for single image super-resolution. In *CVPRW*, pages 136–144, 2017.

[17] David G Lowe. Object recognition from local scale-invariant features. In *ICCV*, volume 2, pages 1150–1157. Ieee, 1999.

[18] Roey Mechrez, Itamar Talmi, Firas Shama, and Lihi Zelnik-Manor. Maintaining natural image statistics with the contextual loss. In *ACCV*, pages 427–443. Springer, 2018.

[19] Roey Mechrez, Itamar Talmi, and Lihi Zelnik-Manor. The contextual loss for image transformation with non-aligned data. In *ECCV*, pages 768–783, 2018.

[20] Ben Niu, Weilei Wen, Wenqi Ren, Xiangde Zhang, Lianping Yang, Shuzhen Wang, Kaihao Zhang, Xiaochun Cao, and Haifeng Shen. Single image super-resolution via a holistic attention network. In *ECCV*, pages 191–207. Springer, 2020.

[21] Taesung Park, Ming-Yu Liu, Ting-Chun Wang, and Jun-Yan Zhu. Semantic image synthesis with spatially-adaptive normalization. In *CVPR*, pages 2337–2346, 2019.

[22] Mehdi SM Sajjadi, Bernhard Scholkopf, and Michael Hirsch. Enhancenet: Single image super-resolution through automated texture synthesis. In *ICCV*, pages 4491–4500, 2017.

[23] Gyumin Shim, Jinsun Park, and In So Kweon. Robust reference-based super-resolution with similarity-aware deformable convolution. In *CVPR*, pages 8425–8434, 2020.

[24] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014.

[25] Libin Sun and James Hays. Super-resolution from internet-scale scene matching. In *2012 IEEE International Conference on Computational Photography (ICCP)*, pages 1–12. IEEE, 2012.

[26] Ying Tai, Jian Yang, and Xiaoming Liu. Image super-resolution via deep recursive residual network. In *CVPR*, pages 3147–3155, 2017.

[27] Dmitry Ulyanov, Andrea Vedaldi, and Victor Lempitsky. Instance normalization: The missing ingredient for fast stylization. *arXiv preprint arXiv:1607.08022*, 2016.

[28] Xintao Wang, Ke Yu, Shixiang Wu, Jinjin Gu, Yihao Liu, Chao Dong, Yu Qiao, and Chen Change Loy. Esrgan: Enhanced super-resolution generative adversarial networks. In *ECCV*, pages 0–0, 2018.

[29] Yi Wang, Xin Tao, Xiaojuan Qi, Xiaoyong Shen, and Jiaya Jia. Image inpainting via generative multi-column convolutional neural networks. In *NeurIPS*, pages 331–340, 2018.

[30] Fuzhi Yang, Huan Yang, Jianlong Fu, Hongtao Lu, and Baining Guo. Learning texture transformer network for image super-resolution. In *CVPR*, pages 5791–5800, 2020.

[31] Yulun Zhang, Kunpeng Li, Kai Li, Lichen Wang, Bineng Zhong, and Yun Fu. Image super-resolution using very deep residual channel attention networks. In *ECCV*, pages 286–301, 2018.

[32] Yulun Zhang, Yapeng Tian, Yu Kong, Bineng Zhong, and Yun Fu. Residual dense network for image super-resolution. In *CVPR*, pages 2472–2481, 2018.

[33] Zhifei Zhang, Zhaowen Wang, Zhe Lin, and Hairong Qi. Image super-resolution by neural texture transfer. In *CVPR*, pages 7982–7991, 2019.

[34] Haitian Zheng, Mengqi Ji, Lei Han, Ziwei Xu, Haoqian Wang, Yebin Liu, and Lu Fang. Learning cross-scale correspondence and patch-based synthesis for reference-based super-resolution. In *BMVC*, 2017.

[35] Haitian Zheng, Mengqi Ji, Haoqian Wang, Yebin Liu, and Lu Fang. Crossnet: An end-to-end reference-based super resolution network using cross-scale warping. In *ECCV*, pages 88–104, 2018.

[36] Xizhou Zhu, Han Hu, Stephen Lin, and Jifeng Dai. Deformable convnets v2: More deformable, better results. In *CVPR*, pages 9308–9316, 2019.