Boosting High-Level Vision with Joint Compression Artifacts Reduction and Super-Resolution

Xiaoyu Xiang
Purdue University
West Lafayette, IN 47907, USA
xiang43@purdue.edu

Qian Lin
HP Labs
Palo Alto, CA 94304, USA
qian.lin@hp.com

Jan P. Allebach
Purdue University
West Lafayette, IN 47907, USA
allebach@purdue.edu

***Abstract*—Due to the limits of bandwidth and storage space, digital images are usually down-scaled and compressed when transmitted over networks, resulting in loss of details and jarring artifacts that can lower the performance of high-level visual tasks. In this paper, we aim to generate an artifact-free high-resolution image from a low-resolution one compressed with an arbitrary quality factor by exploring joint compression artifacts reduction (CAR) and super-resolution (SR) tasks. First, we propose a context-aware joint CAR and SR neural network (CAJNN) that integrates both local and non-local features to solve CAR and SR in one-stage. Finally, a deep reconstruction network is adopted to predict high quality and high-resolution images. Evaluation on CAR and SR benchmark datasets shows that our CAJNN model outperforms previous methods and also takes 26.2% less runtime. Based on this model, we explore addressing two critical challenges in high-level computer vision: optical character recognition of low-resolution texts, and extremely tiny face detection. We demonstrate that CAJNN can serve as an effective image preprocessing method and improve the accuracy for real-scene text recognition (from 85.30% to 85.75%) and the average precision for tiny face detection (from 0.317 to 0.611).**

I. Introduction

Image down-scaling and compression techniques are widely used to meet the limits of hardware storage and data capacity, which sometimes sacrifice the visual effects as well as bringing troubles to visual detection and recognition. Compression artifact reduction (CAR) and single image super-resolution (SISR) [1] have been used in manifold applications, *e.g.* digital zoom on smartphones [2], video streaming [3] and print quality enhancement [4], [5] to restore a high-quality and high-resolution image. Since Dong [6] first proposed SRCNN that applied a three-layer convolutional neural network (CNN) for the SISR task, more and more works [7], [8], [9] have explored how to make use of the deep neural networks (DNN) to achieve better image quality as measured by PSNR and SSIM [10], or better visual quality [11], [12] as measured by other perceptual metrics [13], [14].

Conventional methods adopt a two-stage pipeline to leverage the quality and resolution of real-world images: first preprocess the user's photos with a compression artifacts reduction (CAR) algorithm [17], [18], [19], [20], [21], and then conduct a super-resolution (SR) algorithm [1], [6], [7], [8], [9], [16], [22], [23], [24]. However, the CAR step often causes loss of high-frequency information, which results in a lack of detail in reconstructed SR images. Besides, the computation and data transmission between the two models is time-consuming. To deal with these issues, a single-stage method that jointly solves the Compression Artifact Reduction and Super-Resolution (CARSR) problems is needed to reach a balance between reducing artifacts while retaining most details for the upscale step with a short run-time.

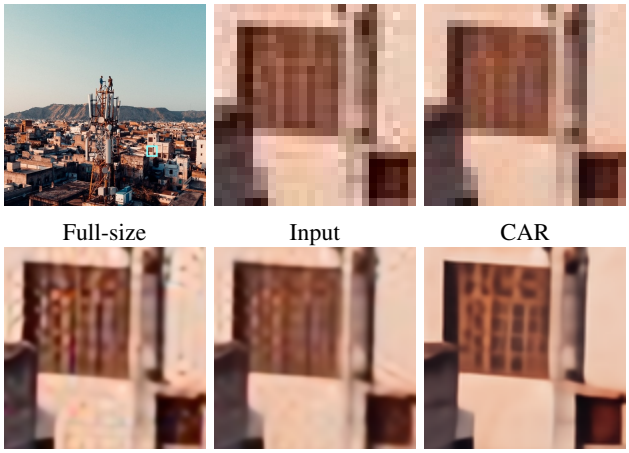


Fig. 1: Demonstration of the joint CAR and SR task. For a user's image without ground truth, our joint CAR and SR model (CAJNN) can generate better output with sharper edges and significantly fewer artifacts compared with either CAR (DnCNN [15]), SR (RCAN [16]), or a two-stage CAR+SR method with the above models.

Both CAR and SR aim to learn the high-frequency information for reconstruction. Thus, instead of simply concatenating two networks together, we can design two functional modules in a single-stage network that reduces the model size by simplifying the two reconstruction processes into one, and can directly obtain high-quality SR output without reconstructing the intermediate artifact-free LR images. Towards this end, we propose a context-aware joint CAR and SR neural network (CAJNN) that can make use of the locally related features in low-quality, low-resolution images to reconstruct high-quality, high-resolution images. To train this network, we construct a paired LR-HR training dataset based on modeling the degradation kernels of web images. Our model turns out to be able to reconstruct high-resolution and artifact-free images with high stability for user's images from a garden variety of web-apps

(*e.g.* Facebook, Instagram, WeChat). Figure 1 illustrates the performance of our proposed algorithm and the benefits of the single-stage joint CAR and SR method compared with previous SR models and two-stage methods: our result can reconstruct a more visually appealing output with accurate structures, sharp edges, and significantly fewer compression artifacts. These output images are not only more recognizable for human viewers, but also for off-the-shelf computer vision algorithms. In this paper, we demonstrate that our proposed CAJNN can enhance the detection and recognition accuracy of high-level vision tasks by reducing the compression artifacts and increasing the resolution of input images.

To summarize, our contributions are mainly three-fold: (1) We propose a novel CAJNN framework that jointly solves the CAR and SR problems for real-world images, that are from unknown devices with unknown quality factors. Here, we explore ways to represent and combine both non-local and local information to enforce image reconstruction performance. Our method doesn't require prior quality factor information of the input. (2) Our experiments show that CAJNN achieves the new state-of-the-art performance on multiple datasets *e.g.* Set5 [25], Set14 [26], BSD100 [27], Urban100[28], *etc.* as measured by the PSNR and SSIM [10] metrics. Compared with the prior art, it generates more stable and reliable outputs for any level of compression quality factors. (3) We provide a new idea for enhancing high-level computer vision tasks like real-scene text recognition and extremely tiny face detection. By preprocessing the input data with our pretrained model, we can improve the performance of existing detectors. Our model demonstrates its effectiveness as a supportive technique on the WIDER face dataset [29] and the ICDAR2013 Focused Scene Text dataset [30].

II. Related Work

CNN-based Single Image Super-Resolution Convolutional Neural Network (CNN) methods have demonstrated a remarkable capability to recover LR images with known kernels after the pioneering work of Dong *et al.* [6] that adopted a 3-layer CNN to learn an end-to-end mapping from LR images to HR images. The follow-up work FSRCNN [23] established the general structure of most SR networks until today, which conducts most computations in the low-resolution domain and upsamples the image to the required scale at the end of the network. After 2016, more and more works began to explore how to make the network go deeper. EDSR [7] reduces the number of parameters by removing the batch normalization layer, and shares the parameters between the low-scale and high-scale models to achieve better training results. RDN [8], [9] and RRDB [12] employ densely-connected residual groups as the major reconstruction block to reach large depth and to allow sufficient low-frequency information to be bypassed. In the meantime, some useful structures have been introduced to enhance the processing speed or output quality. Shi *et al.* [24] designed a sub-pixel upscaling mechanism. RCAN [16] introduces a channel attention mechanism to rescale channel-wise features adaptively, and SAN [31] exploits a more powerful feature expression with second-order channel attention.

Compression Artifacts Reduction Lossy compression methods are widely applied in web image transmission due to their higher compression rates. Traditional methods for the CAR problem generally fall into two categories: unsupervised methods, which include removing noise and increasing sharpness [19], and supervised methods like dictionary-based algorithms [32]. After the success of SRCNN on the super-resolution task, Yu *et al.* directly applied its architecture to compression artifacts suppression. Similar to the development of SR, CNN-based CAR networks can also change from shallow to deep with the introduction of residual blocks and skip connections [33], [15], [34]. Besides, SSIM loss is also employed [18] as a supervision method to obtain better performance than MSE loss. JPEG-related priors are also considered in the network structure design, *e.g.* DDCN [35] adds a Discrete Cosine Transform (DCT)-domain before the dual networks, and the D^3 method [36] takes a further step in the practice of dual-domain approaches [32] by converting sparse-encoding approaches into a one-step sparse inference module.

Unlike the above approaches that require recovering the intermediate LR images with reduced artifacts, our joint CARSR framework directly obtains artifact-free HR images without prior information of quality factors or explicit supervision of CAR in the intermediate LR domain.

III. Joint Compression Artifacts Reduction and Super-Resolution

Given an LR JPEG-compressed image I^{LRLQ}, our goal is to reconstruct the high resolution, high-quality image $G(I^{LRLQ})$ that approaches the high-resolution, high-quality ground truth I^{HRHQ} with a generator G. The CARSR task can be expressed as:

$$\underset{\theta}{\operatorname{argmin}}\, l(I^{HRHQ}, G(I^{LRLQ}, \theta_g)), \tag{1}$$

where l is any designated loss function (*e.g.* MSE, L1, Charbonnier, *etc.*). G is the function representing our deep neural network with parameters θ, for which we wish $G_\theta \approx F^{-1}((I^{LRLQ} \otimes k) \uparrow_s, q)$, where $\otimes$ stands for the convolution operation, k is the degradation kernel of downsampling (*e.g.* bicubic), and s is the downscaling factor. To effectively handle the CARSR task, we propose a single-stage framework, CAJNN. Our proposed model is end-to-end trainable with I^{HRHQ} and I^{LRLQ} pairs according to the function above.

The CAJNN framework mainly consists of three modules (see Figure 2): the *context-aware feature extractor*, the *reconstruction module*, and the *upsampling and enhancement module*. The context-aware feature extractor captures and assembles both intra- and inter-block information with filters of different receptive fields. The reconstruction module further refines the extracted feature maps. Finally, after the processing of the upsampling and enhancement module, these feature maps are converted to high-resolution outputs.

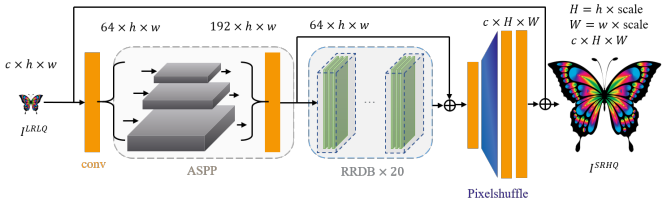


Fig. 2: The network architecture of our proposed CAJNN. It directly reconstructs artifact-free HR images from the LR low-quality images I^{LRLQ}. Atrous Spatial Pyramid Pooling (ASPP) is adopted to utilize the inter-block features and intra-block contexts for the joint CARSR task. The reconstruction module turns the features into a deep feature map, which is converted to a high-quality SR output I^{SRHQ} by the upsampling and enhancement module.

A. Model

Here we discuss the CAJNN structure in detail. The majority of our network operates in the feature domain. Given I^{LRLQ} ($c \times h \times w$ in size), a feature extraction layer first turns the image into feature maps ($n_f \times h \times w$ in size, where n_f denotes the number of feature channels) in the domain for the following process. The feature map will be converted to a high-resolution image ($c \times sh \times sw$ in size) after passing the upsampling and enhancement module. To achieve a balance between GPU capacity and output quality, we apply 64 channels to ensure that enough information is included for the reconstruction computation. We adopt a 3×3 convolution layer that serves as the initial feature extractor. After this module, the input image I^{LRLQ} is turned into a $64 \times h \times w$ tensor f^L.

1) Context-aware Feature Extractor: The pipeline of JPEG compression involves the following steps: color space transformation (*e.g.* JPEG, H.264/AVC, H.265/HEVC), downsampling, block splitting, discrete cosine transform (DCT), quantization, and entropy encoding. Some previous research assumed that the quality factors of input images are known, and the original images are well-aligned by 8×8 patches with respect to the JPEG block boundaries. However, real-world inputs cannot always satisfy such assumptions. In the worst case, the input images might be compressed multiple times and contain sub-blocks or larger blocks, which requires the model to be insensitive, or even blind to the encoding block alignment. Thus, in practice, the spatial context information of both intra- and inter- JPEG blocks is essential for designing a CARSR network.

Since the JPEG block is 8×8 pixels in size, as mentioned before, both intra- and inter-block information need to be taken into consideration in designing the joint CAR and SR network. Thus, we adopt the ASPP module to extract and integrate multi-scale features with an atrous spatial pooling pyramid (ASPP) [37]. We adjust the dilation rates of each layer in the pyramid to extend the filter's perceptive field for extracting different ranges of context information, in which the largest field-of-view should cover the 8×8 block. Besides, we should avoid sampling overlap in different levels of the 3×3 convolutions. Considering the factors above, we choose 1, 3, 4 as the dilation groups to find a good balance between accurately retrieving local details and assimilating context information between adjacent blocks. The input tensors are sent to 3 layers of the pyramid individually: a 3×3 convolutional layer with dilation rate = 1, a 3×3 convolutional layer with dilation rate = 3, and a 3×3 convolutional layer with dilation rate = 4. The outputs of these three layers are concatenated and aggregated by a 1×1 convolution. The process in ASPP can be described by:

$$f^{L'} = [C_{3\times3,1} \otimes f^L | C_{3\times3,3} \otimes f^L | C_{3\times3,4} \otimes f^L] \otimes C_{1\times1,1}, \quad (2)$$

where $f^{L'}$ denotes the output feature ($64 \times h \times w$ in size), $C_{a\times a,r}$ represents the parameters of $a \times a$ convolution with dilation rate r, and $|$ is a concatenation operation.

2) Reconstruction: RRDB (residual-in-residual dense block) [12] is applied as the basic block for the reconstruction trunk. Compared with residual blocks, it densely connects the convolution layers to local groups while removing the batch normalization layer. In our network, the reconstruction module includes 20 RRDBs.

3) Upsampling and Enhancement: After the reconstruction module, the image feature is preprocessed by a 3×3 convolution layer before sending it to the PixelShuffle layer [24] for upsampling. The Pixelshuffle layer produces an HR image from LR feature maps directly with one upscaling filter for each feature map. Compared with the upconvolution, the PixelShuffle layer is $\log_2 s^2$ times faster in theory because of applying sub-pixel activation to convert most of the computations from the HR to the LR domain. The feature $f^{L''}$ is turned into a $c \times sh \times sw$ HR image by the PixelShuffle layer, which can be described by:

$$I^{SR'} = PS(W_L \otimes f^{L''} + b_L), \quad (3)$$

where W_L denotes the convolution weights and b_L the bias in the LR domain, PS is a periodic shuffling operator for rearranging the input LR feature tensor $f^{L''}$ ($c \times s^2 \times h \times w$) to a HR tensor of shape $c \times rh \times rw$:

$$PS(T)_{x,y,c} = T_{\lfloor x/s \rfloor, \lfloor y/s \rfloor, c\cdot s\cdot \mathrm{mod}(y,s)+c\cdot s\cdot \mathrm{mod}(x,s)}. \quad (4)$$

Instead of directly outputting the high-resolution image, we process it through two 3×3 convolution layers for further enhancement, and get $I^{SR} = C_{3\times3,1} \otimes (C_{3\times3,1} \otimes I^{SR'})$.

To make the major network focus on learning the high-frequency information in the input image, we bilinearly upsample the input LR image I^{LRLQ} and add it to form the final output $G(I^{LRLQ}, \theta_g)$:

$$G(I^{LRLQ}, \theta_g) = I^{LRLQ} \uparrow_s + I^{SR}. \quad (5)$$

This long-range skip connection changes the target of our major network from directly reconstructing a high-resolution image to reconstructing its residual. By letting the low-frequency information of the input bypass the major network, it lowers the difficulty of reconstruction and promotes the convergence speed of the network.

IV. Experiments and Analysis

A. Experimental Setup

Training Dataset In this paper, we choose the DIV2K dataset [38] (800 RGB images of 2k resolution) and Flick2K dataset [39] (2650 RGB images of 2k resolution) as our training set. To get the training pairs that approach the degradation kernels of web images, we first model the downsampling and compression types of popular web platforms. Besides, we also discover that adding severely compressed samples to the training set can improve the output quality in terms of PSNR, even for input images compressed with high-quality factors. Based on these pre-experimental results, the images of the training set are downsampled with a scaling factor $s = 4$ and compressed by MATLAB [40] with random quality factors from 10 to 100. Besides, we perform data augmentation on these images by randomly cropping, randomly rotating by $90°$, $180°$, and $270°$, and randomly horizontal-flipping. As a result, each cropped image patch can have eight different positions at maximum.

Test Datasets We compare the performance of our model and previous methods on Set5 [25], Set14 [26], BSD100 [27], Urban100 [28] and Manga109 [41]. Each image is downscaled by the scaling factor $s = 4$ relative to the original size and compressed with quality factors of 10, 20 and 40 to be consistent with previous works.

Implementation Details Our network is trained on one Nvidia Titan Xp graphics card. The batch size is 36, and the patch size is 128 for ground truth and 32 for low-resolution input. We use Adam [42] as the optimizer with a cosine annealing learning rate, in which the initial learning rate is $2e-4$, and the minimum learning rate is $1e-7$. The scheduler restarts every 250,000 iterations. We trained the network for 1,000,000 iterations in total.

B. Results for Image Quality Assessment

We report the PSNR and SSIM [10] of the Y channels in the test sets to be consistent with previous works.

Comparison with SOTA on Standard Test Sets We compare the performance of CAJNN to the previous state-of-the-art (SOTA) methods on the standard test sets as mentioned above. In addition to PSNR and SSIM, we also show the number of parameters and runtime (the inference time on Set5) in Table I. Depending on the workflow for solving the CAR and SR problem, these methods can be categorized into the following three types: (1) *SR:* directly use pretrained SR models. (2) *CAR+SR:* the aforementioned two-stage method, which first removes the compression artifacts and then sends the output images to the SR model. (3) *Joint CAR & SR:* the single-stage method that jointly handles CAR and SR with one model. We report both the direct output and the self-ensembled output of our network.

As can be seen in Table I, CAJNN significantly outperforms the existing methods at different quality factors, improving the PSNR for all QFs, and yielding the highest overall PSNR for Set5. The improvement is consistently observed on SSIM, as well. Moreover, our model is more light-weight than most of the current models, including one-stage and two-stage in summation, which results in faster inference speed on the same hardware (all the tests are conducted on one Nvidia Titan Xp graphics card).

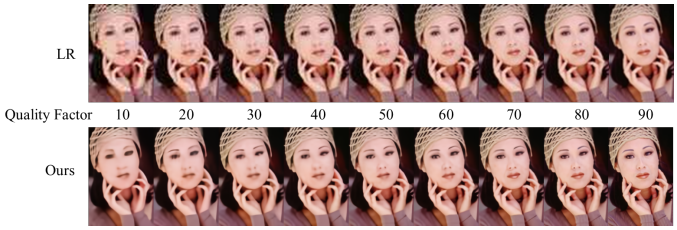


Fig. 3: The qualitative result of our network from compressed images with different quality factors (zoom in for a better view). Our model is able to reconstruct reasonable SR images, even at extremely low quality factors. Besides, our results are free of color jittering and other inconsistencies for such a wide range of compression ratios. The image is the "woman" image from Set5 [25]

Figure 3 gives a qualitative example of the result of our model, where the input image is *woman* from the Set5 [25] that is downsampled and compressed by a wide range of quality factors from 10 to 100. It is worth noting that compression with very low quality factors causes a significant color shift on the hue and spatial distribution of the original image, which can be seen in the leftmost LR image (QF = 10). Our model is able to correctly restore the color aberrations of RGB images with a high consistency among different QFs.

Results on User Images Besides the above experiments on standard test images, we also conduct experiments on real user images to demonstrate the effectiveness of our model. We mainly focus on the perceptual effect since there are no ground-truth images. Figure 4 shows the CAJNN results on real-world image from the WIDER face dataset [29]. For comparison, RCAN[16] and RRDB [12] are used as representative SR method, ARCNN [17] and DnCNN [15] are used as the representative CAR methods. The real-world images have unknown downsampling kernels and compression mechanisms, depending on the platforms. According to Figure 4, the SR methods generate images with obvious color shift and ringing artifacts. These artifacts are alleviated with two-stage methods. Still, the results are blurry. Compared with the two-stage methods, our CAJNN can provide SR outputs with sharp edges and rich details, which demonstrates the superiority of our proposed single-stage method when applied to real-world CARSR problems.

C. Results for Low-Resolution Text Recognition

Comparing the input LR image and our output in Figure 4, the texts become more readable after being processed by our model. Inspired by this observation, we conducted the following experiments to explore our model's potential to leverage real-scene text recognition task for low-resolution characters.

TABLE I: Quantitative comparison of applying SOTA SR methods, two-stage SR and CAR methods, and our CAJNN. The best two results are highlighted in red and blue colors, respectively. Our method greatly outperforms all two-stage methods in terms of PSNR and SSIM, while having a relatively small model size and less runtime.

QF	Method	Network	Runtime (s)	Parameters (Million)	Set5		Set14		BSD100		Urban100		Manga109	
					PSNR	SSIM	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM
10	SR	Bicubic	-	-	23.99	0.6329	22.94	0.5513	23.33	0.5303	20.95	0.5182	21.94	0.6383
		EDSR	1.94	43.1	23.41	0.6019	22.48	0.5272	22.96	0.5098	20.57	0.5006	21.53	0.6151
		RCAN	2.04	16	23.14	0.5733	22.29	0.5064	22.78	0.4984	20.36	0.4819	21.21	0.5878
		RRDB	0.65	16.7	22.43	0.5223	22.86	0.5051	20.43	0.4940	20.43	0.4940	21.34	0.6075
	CAR+SR	ARCNN+RRDB	3.20+0.65	0.56+16.7	24.21	0.6699	23.38	0.5774	23.63	0.5474	21.28	0.5466	22.36	0.6856
		DnCNN+RRDB	0.38+0.65	0.06+16.7	24.07	0.6434	23.13	0.5582	23.37	0.5324	21.04	0.5305	22.10	0.6532
	Joint CAR&SR	CAJNN (ours)	0.48	14.8	25.04	0.7169	23.95	0.6028	23.84	0.5598	21.97	0.5977	23.29	0.7333
		CAJNN (ours, self-ensembled)	2.50	14.8	25.14	0.7202	24.03	0.6052	23.88	0.5610	22.18	0.6051	23.44	0.7377
20	SR	Bicubic	-	-	25.32	0.6761	23.85	0.5870	24.14	0.5611	21.66	0.5526	22.84	0.6724
		EDSR	1.94	43.1	24.76	0.6490	23.59	0.5707	23.88	0.5482	21.38	0.5427	22.58	0.6549
		RCAN	2.04	16	24.44	0.6226	23.40	0.5502	23.65	0.5351	21.12	0.5234	22.14	0.6253
		RRDB	0.65	16.7	24.65	0.6450	23.57	0.5661	23.79	0.5442	21.25	0.5365	22.38	0.6474
	CAR+SR	ARCNN+RRDB	3.20+0.65	0.56+16.7	25.40	0.7082	24.30	0.6091	24.39	0.5755	22.02	0.5811	23.52	0.7172
		DnCNN+RRDB	0.38+0.65	0.06+16.7	25.55	0.6946	24.24	0.6001	24.28	0.5679	21.90	0.5732	23.24	0.6961
	Joint CAR&SR	CAJNN (ours)	0.48	14.8	26.59	0.7604	25.03	0.6391	24.70	0.5924	23.06	0.6482	24.81	0.7783
		CAJNN (ours, self-ensembled)	2.50	14.8	26.65	0.7633	25.10	0.6404	24.74	0.5936	23.28	0.6550	24.98	0.7820
40	SR	Bicubic	-	-	26.38	0.7154	24.55	0.6201	24.77	0.5898	22.26	0.5877	23.66	0.7081
		EDSR	1.94	43.1	26.01	0.6972	24.48	0.6120	24.62	0.5836	22.18	0.5893	23.73	0.7003
		RCAN	2.04	16	25.70	0.6726	24.30	0.5936	24.36	0.5704	21.86	0.5690	23.13	0.6673
		RRDB	0.65	16.7	25.99	0.6958	24.50	0.6079	24.54	0.5804	22.10	0.5851	23.50	0.6918
	CAR+SR	ARCNN+RRDB	3.20+0.65	0.56+16.7	26.65	0.7495	25.16	0.6424	25.06	0.6053	22.82	0.6235	24.68	0.7578
		DnCNN+RRDB	0.38+0.65	0.06+16.7	26.87	0.7403	25.15	0.6373	25.00	0.5995	22.78	0.6194	24.42	0.7404
	Joint CAR&SR	CAJNN (ours)	0.48	14.8	28.05	0.7981	25.96	0.6729	25.43	0.6240	24.09	0.6962	26.25	0.8177
		CAJNN (ours, self-ensembled)	2.50	14.8	28.16	0.7993	26.03	0.6742	25.46	0.6251	24.31	0.7011	26.44	0.8211

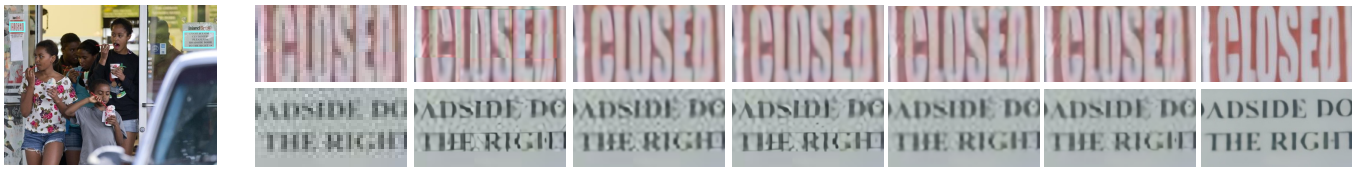


Fig. 4: CAR & SR performance comparison of different methods on a user's image from the WIDER face dataset [29]. Compared with previous methods, our model can generate artifact-free high-resolution images with sharp edges.

We compare the total accuracy of generic text detection on the ICDAR2013 Focused Scene Text dataset [30] with TPS-ResNet-BiLSTM-Attn [43] as the text recognition method. The baseline result is acquired by directly recognizing the original input images. As a comparison with the baseline, we use the CAJNN model as described in previous sections to generate artifact-free SR images from the original images and conduct recognition on the output images.

As can be seen in Table II, the preprocessing of CAJNN improves the recognition accuracy from 85.30% to 85.75%, which indicates that the outputs of our model are not only visually appealing to human viewers, but also include more distinct information for the text recognition network as shown in Figure 5. It is worth noting that our output image is $4\times$ the size compared with the baseline inputs, and the average detection time is increased from 31.22s to 41.56s. Although the improvement in accuracy demonstrates the positive effect yielded by our model, the rise in computation is hard to ignore. Therefore, we disentangle the influence of SR and CAR by bicubicly downsampling the CARSR output images and acquire the third recognition result. Since the image size remains the same as that of the original image, the detection time is identical to the baseline. The recognition accuracy still improves 0.27% compared with the baseline due to the reduction of compression artifacts, which indicates that our model is capable of extracting and maintaining critical features of input images. This experimental result points out a plausible direction for future text recognition research: the image quality plays a vital role in the recognition accuracy, which can be improved by utilizing the priors learned from a pretrained CARSR model.

TABLE II: Text recognition accuracy on the ICDAR 2013 Focused Scene Text dataset [30]. Compared with the baseline method, the introduction of our CARSR method improves the detection performance by 0.45% (without downsampling) and 0.27% (with downsampling).

Method	Accuracy	Detection Time (s)
Baseline [43]	85.30%	31.22
Ours + Baseline [43]	**85.75%**	41.56
Ours + Downsample + Baseline [2]	85.57%	31.22

D. Results for Extremely Tiny Face Detection

Extremely tiny face detection is another practical, yet challenging task in high-level computer vision. Most of the state-of-the-art (SOTA) face detectors [44], [45] for in-the-wild images have already taken various scales and distortions into consideration to achieve impressive detection performance. [46] proposed a solution to tackle tiny face detection by explicitly restoring an HR face from a small blurry one using a Generative Adversarial Network (GAN) [47].

We experimentally validate the effect achieved by our CAJNN on tiny face images in the WIDER FACE dataset [29] by

GT Ours ($\times 4$) Ours (downsampled)

Fig. 5: Test samples of ICDAR2013 dataset [30] (word_161, word_836). The first column shows original input images, the second column is the CARSR output generated by our method, and the third column is acquired by downsampling the second column. By comparing the detection results in the first and second columns, our method can serve as a supportive method for the recognition of low-resolution texts. Besides, the artifact-free image in the third column can also provide more recognizable features for the baseline model without increasing the image size.

TABLE III: Average precision of three data types in the WIDER FACE validation set [29] with the same face detector [48]. The application of our CARSR method greatly improves the detection performance with LR images on all three subsets.

Input Data	Easy	Medium	Hard
GT	0.900	0.887	0.792
LR	0.824	0.692	0.317
LR + Ours	0.893	0.857	0.611

comparing the detection results from the following three types of data: original HR (serves as the baseline), downsampled LR (serves as the extremely tiny face inputs), and CARSR outputs from our model. [48] is applied as the backbone face detector (We use an unofficial PyTorch implementation provided by https://github.com/varunagrawal/tiny-faces-pytorch).

Table III shows the Average Precision (AP) of the downsampled tiny images and our enhanced ones on all the three validation sets (easy, medium, and hard) of WIDER FACE [29]. From Table III, we observe that the data processed by CAJNN dramatically improves the detection of LR inputs from 0.317 to 0.611 in AP on the hard set. The reason is that the baseline detector performs downsampling operations by large strides on the tiny faces. Considering the fact that the tiny faces themselves contain less information than average, the detailed information of face structure is lost after several downsampling convolutions. In contrast, our CAJNN provides an artifact-free SR image, which can boost the detection performance by better utilizing the information of small faces. In Figure 6, the precision-recall curve of our reconstructed image (green line) is close to the ground truth (red line) on the easy and medium subsets. In the hard subset, our CAJNN yields a significant improvement compared to the LR curve. The gap between our output and the GT is due to the irreversible loss of information in extremely tiny faces that happens more frequently in the hard set during the downsampling process.

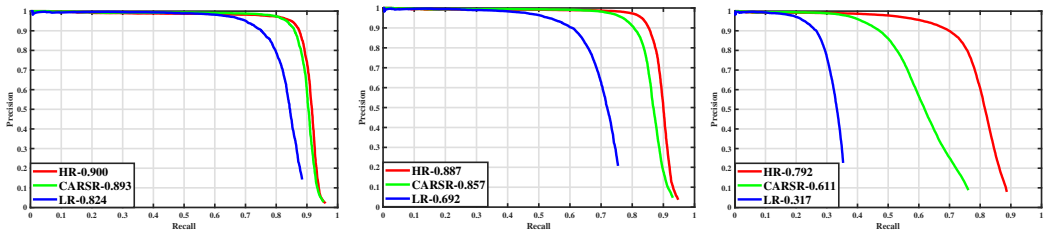
TABLE IV: Ablation Study on the validation set (Set5). We report the performance of CAJNN without the long-range skip connection and ASPP as the baseline. Rows 1-3 show the influence of different ways to extract contextual information by replacing ASPP with other network structures. Rows 4-5 compare the effect of two different upsampling methods on PSNR. The combination of the ASPP and Pixelshuffle modules yields the best performance, and thus is adopted in our network architecture.

Model	Base	1	2	3	4
Non-local module		√			
ASPP			√	√	
Seqeuntial atrous pooling					√
Upconvolution	√	√	√		
Pixelshuffle				√	√
PSNR (dB)	27.868	28.274	28.276	**28.292**	28.262

E. Ablation Study

Effect of Multi-scale Information As discussed in previous sections, both intra- and inter-block context information is important for designing a CARSR network. In other low-level vision tasks, context information at different scales has already been proved to be effective in improving network performance. Inspired by the first convolution layer of the ResNet [49], previous researchers [50] applied 7×7 convolution to extract the context features for the video frame interpolation task. However, such big kernels bring a tremendous number of parameters to the network, especially when embedded in the feature domain, resulting in higher computational cost. Another way of enlarging the filter's receptive field is to use a non-local module [51], [52], where the input images are downsampled by convolutional strides and processed at different scales. The non-local module has a rather complex structure and also a large number of parameters. In order to use the context information in a much simpler and lighter representation, our method adopts atrous convolution. By adjusting the dilation rate r, the filter can incorporate the context information from a larger receptive field without dramatically increasing the number of parameter as compared to the above methods.

We conduct an ablation study to illustrate the effect of different ways of representing contextual information in Table IV. In Rows 1-3, we compare the performance of the non-local module, ASPP, and sequential atrous pooling. Comparing the base model to Column 1 in Table IV, we can conclude that the introduction of multi-scale information via a non-local module can significantly improve the PSNR by 0.406 dB. This result validates the superiority of aggregating both intra- and inter-block features rather than using a purely local representation for the CARSR task. Furthermore, as seen by comparing Columns 1 and 2, replacing the non-local module by our well-designed ASPP can improve the PSNR by 0.002 dB. Although the improvement is rather small, it is worth noting that the ASPP has fewer convolution layers and parameters, which results in a smaller model size and fewer FLOPs. Remarkably, it can achieve results that are comparable, or even



Easy Medium Hard

Fig. 6: The precision-recall curve of three subsets in WIDER FACE [29]. The AOC (area under curve) reflects the detector's performance on each type of data (GT, LR and CARSR). With preprocessing by our model, the detection performance of tiny images can be improved close to that achieved with GT. (Zoom in for a better view.)

better than that yielded by models with more parameters. By comparing Columns 3 and 4, we also note that the PSNR of ASPP is higher than that of sequential atrous pooling by 0.03 dB, which means that the pyramid-fusion structure is more efficient in representing the multi-scale information. Finally, by comparing Columns 2 and 3 of Table IV, we can observe that the PixelShuffle layer brings a 0.16 dB improvement to PSNR.

End-to-End Supervision by Joint CAR and SR Another ablation study on supervising the CARSR task is conducted to illustrate the effect of joint end-to-end training. Instead of supervising with I^{HRHQ}, we attempt to disentangle the CAR and SR by introducing a reconstruction loss according to the definition in Equation 6, where we can generate an artifact-free LR image I^{LRHQ} from the ground truth I^{HRHQ}:

$$I^{LRHQ} = (k \otimes I^{HRHQ}) \downarrow_s, \tag{6}$$

and use it to explicitly supervise the intermediate CAR output $\hat{G}(f^{L'})$ after the context-aware module:

$$l^{LR} = l(I^{LQHQ}, \hat{G}(f^{L'})). \tag{7}$$

Denoting the pixel-wise loss of the final output and ground truth (shown in Equation 1) as l_{HR}, the overall training loss becomes:

$$l = l^{HR} + \lambda l^{LR}, \tag{8}$$

by increasing the weight λ, we can acquire models trained with higher disentanglement levels. We train three models with $\lambda = 0, 1, 16$ while keeping all the other factors the same. The performance of these models on our validation set is shown in Table V. The trend is obvious: the PSNR increases as the entanglement increases, which demonstrates the effectiveness of the joint CARSR method with a single-stage network.

TABLE V: Ablation Study on joint end-to-end supervision. We introduce the explicit reconstruction loss as a disentanglement mechanism of CAR and SR. By changing the weight of this loss term, we can study the effect of different levels of joint-supervision. Among all the settings, the model trained without the reconstruction loss performs best on our validation set.

Model	a	b	c
Weight of reconstruction loss λ	16	1	0
PSNR (dB)	27.507	27.627	**27.672**

V. Conclusion

In this paper, we propose a single-stage network for the joint CARSR task to directly reconstruct an artifact-free high-resolution image from a compressed low-resolution input. To address the CARSR problem, we make use of the contextual information by introducing a specially designed ASPP that integrates both intra- and inter-block features. Our experiments illustrate the effectiveness and efficiency of our method with both standard test images and real-world images. Moreover, the extensive experimental results reveal a high potential of enhancing the performance of current methods for various high-level computer vision tasks, *e.g.* real-scene resolution text recognition, and extremely tiny face detection.

Acknowledgment

This research is supported by HP Inc., Palo Alto, CA.

References

[1] J. Allebach and P. W. Wong, "Edge-directed interpolation," in *Proceedings of 3rd IEEE International Conference on Image Processing*, vol. 3. IEEE, 1996, pp. 707–710.

[2] B. Wronski, I. Garcia-Dorado, M. Ernst, D. Kelly, M. Krainin, C.-K. Liang, M. Levoy, and P. Milanfar, "Handheld multi-frame super-resolution," *arXiv preprint arXiv:1905.03277*, 2019.

[3] X. Xiang, Y. Tian, Y. Zhang, Y. Fu, J. P. Allebach, and C. Xu, "Zooming slow-mo: Fast and accurate one-stage space-time video super-resolution," *arXiv preprint arXiv:2002.11616*, 2020.

[4] Z. Xiao, M. Nguyen, E. Maggard, M. Shaw, J. Allebach, and A. Reibman, "Real-time print quality diagnostics," in *Image Quality and System Performance XIV (Part of IS&T Electronic Imaging 2017)*, no. 12. Society for Imaging Science and Technology, 2017, pp. 174–179.

[5] X. Xiang, R. Jessome, E. Maggard, Y. Bang, M. Cho, and J. Allebach, "Blockwise based detection of local defects," in *Image Quality and System Performance XVI (Part of IS&T Electronic Imaging 2019)*, no. 10. Society for Imaging Science and Technology, 2019, pp. 303–1.

[6] C. Dong, C. C. Loy, K. He, and X. Tang, "Learning a deep convolutional network for image super-resolution," in *Proceedings of the European Conference on Computer Vision*. Springer, 2014, pp. 184–199.

[7] B. Lim, S. Son, H. Kim, S. Nah, and K. Mu Lee, "Enhanced deep residual networks for single image super-resolution," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, 2017, pp. 136–144.

[8] Y. Zhang, Y. Tian, Y. Kong, B. Zhong, and Y. Fu, "Residual dense network for image super-resolution," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 2472–2481.

[9] ——, "Residual dense network for image restoration," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2020.

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

[11] C. Ledig, L. Theis, F. Huszár, J. Caballero, A. Cunningham, A. Acosta, A. Aitken, A. Tejani, J. Totz, Z. Wang *et al.*, "Photo-realistic single image super-resolution using a generative adversarial network," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017, pp. 4681–4690.

[12] X. Wang, K. Yu, S. Wu, J. Gu, Y. Liu, C. Dong, Y. Qiao, and C. C. Loy, "Esrgan: Enhanced super-resolution generative adversarial networks," in *The European Conference on Computer Vision Workshops*, September 2018.

[13] J. Johnson, A. Alahi, and L. Fei-Fei, "Perceptual losses for real-time style transfer and super-resolution," in *Proceedings of the European Conference on Computer Vision*. Springer, 2016, pp. 694–711.

[14] C. Ma, C.-Y. Yang, X. Yang, and M.-H. Yang, "Learning a no-reference quality metric for single-image super-resolution," *Computer Vision and Image Understanding*, vol. 158, pp. 1–16, 2017.

[15] P. Svoboda, M. Hradis, D. Barina, and P. Zemcik, "Compression artifacts removal using convolutional neural networks," *arXiv preprint arXiv:1605.00366*, 2016.

[16] Y. Zhang, K. Li, K. Li, L. Wang, B. Zhong, and Y. Fu, "Image super-resolution using very deep residual channel attention networks," in *Proceedings of the European Conference on Computer Vision*, 2018, pp. 286–301.

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

[18] L. Galteri, L. Seidenari, M. Bertini, and A. Del Bimbo, "Deep generative adversarial compression artifact removal," in *Proceedings of the IEEE International Conference on Computer Vision*, 2017, pp. 4826–4835.

[19] B. Zhang and J. P. Allebach, "Adaptive bilateral filter for sharpness enhancement and noise removal," *IEEE Transactions on Image Processing*, vol. 17, no. 5, pp. 664–678, 2008.

[20] B. Zhang, J. Gu, C. Chen, J. Han, X. Su, X. Cao, and J. Liu, "One-two-one networks for compression artifacts reduction in remote sensing," *ISPRS journal of Photogrammetry and Remote sensing*, vol. 145, pp. 184–196, 2018.

[21] K. Zhang, W. Zuo, Y. Chen, D. Meng, and L. Zhang, "Beyond a Gaussian denoiser: Residual learning of deep CNN for image denoising," *IEEE Transactions on Image Processing*, vol. 26, no. 7, pp. 3142–3155, 2017.

[22] C. B. Atkins, C. A. Bouman, and J. P. Allebach, "Optimal image scaling using pixel classification," in *Proceedings 2001 International Conference on Image Processing (Cat. No. 01CH37205)*, vol. 3. IEEE, 2001, pp. 864–867.

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

[24] 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 *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2016, pp. 1874–1883.

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

[26] R. Zeyde, M. Elad, and M. Protter, "On single image scale-up using sparse-representations," in *International Conference on Curves and Surfaces*. Springer, 2010, pp. 711–730.

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

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

[29] S. Yang, P. Luo, C. C. Loy, and X. Tang, "WIDER FACE: A face detection benchmark," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016.

[30] D. Karatzas, F. Shafait, S. Uchida, M. Iwamura, L. G. i Bigorda, S. R. Mestre, J. Mas, D. F. Mota, J. A. Almazan, and L. P. De Las Heras, "ICDAR 2013 robust reading competition," in *2013 12th International Conference on Document Analysis and Recognition*. IEEE, 2013, pp. 1484–1493.

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

[32] X. Liu, X. Wu, J. Zhou, and D. Zhao, "Data-driven sparsity-based restoration of JPEG-compressed images in dual transform-pixel domain," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2015, pp. 5171–5178.

[33] H. Chen, X. He, C. Ren, L. Qing, and Q. Teng, "Cisrdcnn: Super-resolution of compressed images using deep convolutional neural networks," *Neurocomputing*, vol. 285, pp. 204–219, 2018.

[34] S. Zini, S. Bianco, and R. Schettini, "Deep residual autoencoder for quality independent JPEG restoration," *arXiv preprint arXiv:1903.06117*, 2019.

[35] J. Guo and H. Chao, "Building dual-domain representations for compression artifacts reduction," in *Proceedings of the European Conference on Computer Vision*. Springer, 2016, pp. 628–644.

[36] Z. Wang, D. Liu, S. Chang, Q. Ling, Y. Yang, and T. S. Huang, "D3: Deep dual-domain based fast restoration of JPEG-compressed images," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2016, pp. 2764–2772.

[37] L.-C. Chen, G. Papandreou, I. Kokkinos, K. Murphy, and A. L. Yuille, "Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 40, no. 4, pp. 834–848, 2017.

[38] E. Agustsson and R. Timofte, "Ntire 2017 challenge on single image super-resolution: Dataset and study," in *The IEEE Conference on Computer Vision and Pattern Recognition Workshops*, July 2017.

[39] R. Timofte, E. Agustsson, L. Van Gool, M.-H. Yang, and L. Zhang, "Ntire 2017 challenge on single image super-resolution: Methods and results," in *The IEEE Conference on Computer Vision and Pattern Recognition Workshops*, 2017, pp. 114–125.

[40] MATLAB, *9.5.0.944444 (R2018b)*. Natick, Massachusetts: The MathWorks Inc., 2018.

[41] Y. Matsui, K. Ito, Y. Aramaki, A. Fujimoto, T. Ogawa, T. Yamasaki, and K. Aizawa, "Sketch-based manga retrieval using manga109 dataset," *Multimedia Tools and Applications*, vol. 76, no. 20, pp. 21 811–21 838, 2017.

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

[43] J. Baek, G. Kim, J. Lee, S. Park, D. Han, S. Yun, S. J. Oh, and H. Lee, "What is wrong with scene text recognition model comparisons? dataset and model analysis," *arXiv preprint arXiv:1904.01906*, 2019.

[44] C. Zhu, R. Tao, K. Luu, and M. Savvides, "Seeing small faces from robust anchor's perspective," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 5127–5136.

[45] Y. Yoo, D. Han, and S. Yun, "Extd: Extremely tiny face detector via iterative filter reuse," *arXiv preprint arXiv:1906.06579*, 2019.

[46] Y. Bai, Y. Zhang, M. Ding, and B. Ghanem, "Finding tiny faces in the wild with generative adversarial network," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 21–30.

[47] 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, pp. 2672–2680.

[48] P. Hu and D. Ramanan, "Finding tiny faces," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017, pp. 951–959.

[49] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2016, pp. 770–778.

[50] S. Niklaus and F. Liu, "Context-aware synthesis for video frame interpolation," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 1701–1710.

[51] D. Liu, B. Wen, Y. Fan, C. C. Loy, and T. S. Huang, "Non-local recurrent network for image restoration," in *Advances in Neural Information Processing Systems*, 2018, pp. 1673–1682.

[52] X. Wang, R. Girshick, A. Gupta, and K. He, "Non-local neural networks," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 7794–7803.