Edge-Aware Image Super-Resolution using a Generative Adversarial Network

Bishshoy Das[1*] and Sumantra Dutta Roy[1]

[1*]Department of Electrical Engineering, Indian Institute of Technology Delhi, Hauz Khas, New Delhi, 110016, Delhi, India.

*Corresponding author(s). E-mail(s): bishshoy.das@ee.iitd.ac.in;
Contributing authors: sumantra@ee.iitd.ac.in;

Abstract

Edge-awareness is an important factor in the perception of high frequency details. MSE-based single image super-resolution (SISR) algorithms, such as SRResNet do not deliver perceptually sharp images, but maximizes PSNR (Peak Signal-to-Noise Ratio). Edge details are often lost in such algorithms. A variant of SRResNet based on a generative adversarial network (GAN) model, named SRGAN, aims at achieving higher perceptual sharpness by trading in PSNR. This drop in PSNR is often massive and is attributed to the occurrence of unwanted artifacts. We introduce EaSRGAN, an edge-aware generative adversarial network, which reduces artifacts, delivers highly sharp and photorealistic images with PSNR values better than SRGAN. EaSRGAN treats high frequency regions separately from flat regions which brings in awareness of edges in the super-resolution output. Combined with a multi-stage training process separate for edge and flat areas, these loss functions make the generator and the discriminator, 'edge-aware'. We compare our results with state-of-the-art SISR algorithms. EaSRGAN delivers superior perceptual clarity like that of SRGAN, while maintaining high PSNR by attenuating artifacts.

Keywords: super-resolution, image enhancement, edge detection, generative adversarial network

Fig. 1: Unlike the 4× upscaling in SRGAN (left), our EaSRGAN output (center) suppresses artifacts to obtain a result close to the original HR image (right). The bottom row shows the zoomed-in yellow box.

1 Introduction

PSNR scores in SRCNN [1] outperform earlier approaches in Single Image Super-Resolution (SISR, hereafter) by a large margin. Other examples of deep learning-based approaches include [2], [3], [4], [5], [6], [7], [8], [9], [10] and [11].

MSE-based approaches learn to produce the expected value of the distribution rather than a plausible sample, hence resulting in smooth outputs. The SRGAN approach [12] is designed to select a plausible sample from the underlying distribution. It is based on perceptual loss [4]. The state-of-the-art MSE-based approach is possibly SRResNet, which forms the generator in SRGAN. In other variants of SRResNet, such as RCAN [13], the higher PSNR comes through the use of Residual-in-Residual blocks and very deep architectures (200+ layers).

Photo-realistic algorithms such as SRGAN [12], ProGANSR [14], EnhanceNet [15], ESRGAN [16] and EPSR[17] have an inadvertant side-effect of producing low PSNR output and typically, visual artifacts as well. The authors in [18] present an edge-based approach by incorporating an edge loss. However, the VGG loss that is used in all of these methods overshadows the edge loss and attenuates its impact. Further, the absence of edge features in the discriminator precludes it from being entirely edge-aware.

Our motivation for this work is to develop a method that preserves the strengths of both SRResNet and SRGAN. We present an 'Edge-aware'

approach: EaSRGAN achieves higher PSNR values than SRGAN, while delivering perceptual sharpness in the Super-Resolution (SR, hereafter) output. We show that Intersection-over-Union (IoU, hereafter) scores of EaSRGAN are higher than SRGAN. They come close to those of SRResNet, in spite of being a photo-realistic algorithm. EaSRGAN uses separate loss functions to reduce artifact intensity in both high and low frequency regions. The algorithm does not suffer from the documented instability issues of SRGAN [12].

Multi-image super-resolution (MISR) is a technique that combines two or more images of the same scene to create a single higher-resolution image. As a result, various sources of data are always accessible for creating the SR image. All of the images that are in the pipeline of an MISR algorithm provides true information about the actual underlying data distribution of the scene.

In single-image super-resolution (SISR), there is only one image accessible at the input, and no other source of information is provided. A neural network trained for SISR, learns the mapping from low resolution to high resolution using learned feature representations from other sources of data, such as a training dataset. However, the true data distribution that is available to an MISR algorithm is unavaliable to an SISR algorithm. A neural network or any prior based algorithm for SISR has to basically guess the distribution of the upscaled scene. Since neural networks are trained on millions of images, the guesses are educated guesses. We call this phenomenon of educated guessing, 'detail hallucination'. Much of the artifacts that are found in generative SISR algorithms such as SRGAN [12] are a result of over-hallucination. Even though artifacts arise as a result of hallucination, details in high frequency regions also arise by the same underlying proces. In section [2.1] and [3.2.1], we perform extensive empirical analysis of SRGAN and show how SRGAN leads to artifacts in both high frequency and flat regions. Our proposed method EaSRGAN identifies the high frequency regions and selectively allows the generetive neural network to hallucinate fine details in those specific regions, all while maintaining smoothness in flat regions. The basis of our proposed method is that artifact production can be controlled by controlling the amount of details a neural network produces across different regions of the image. In EaSRGAN, Higher quality SR (with high PSNR) comes from two ways: controlling the effect of fine detail hallucination by using different loss functions, and containing the extent of the artifacts by marking different regions. Flat regions are easy to upscale. Regions with edges have information outside the Nyquist bound. SRResNet targets MSE minimization and hence, does not render edges in the SR image. This leads to blurry and unsharp output. This also accounts for the lower PSNR in SRGAN [12]. The generator network is pushed to choose possible hypotheses incorporating adversarial loss functions. This results in over-hallucination, and higher perceptual sharpness at the cost of lower PSNR values. We additionally observe that SRGAN alters the best-achieved low-frequency details, to produce low-frequency artifacts. The loss function in SRGAN targets both regions (low- and high-frequency, alike) separately for controlled hallucination. Our EaSRGAN targets these lacunae, as described below.

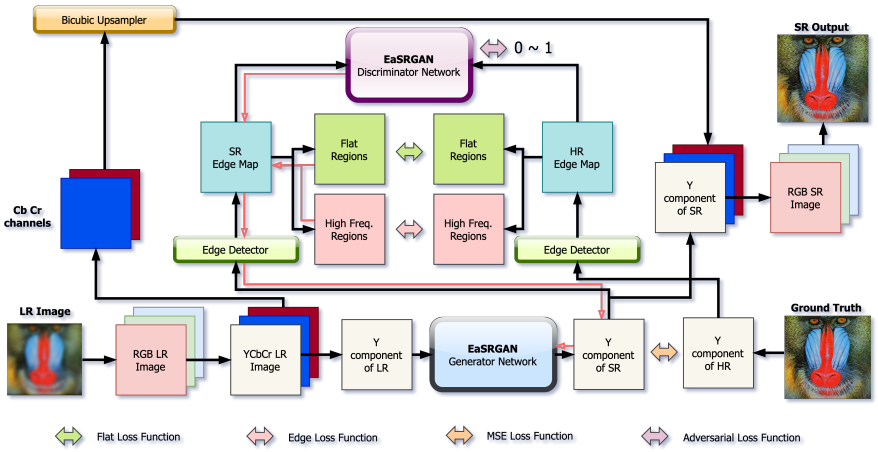


Fig. 2: The EaSRGAN training pipeline: the LR input image is first converted to YCbCr (Luminance (Y), Chroma Blue (Cb), Chroma Red (Cr)). The Cb and Cr channels are upsampled using bicubic interpolation. The Y channel (luma) is passed through the generator network for super-resolution and the super-resolved output (denoted by SR) is obtained. It is then passed to the edge detection module where flat and high frequency regions are separated and compared with corresponding flat and high frequency regions of the ground truth high resolution image (denoted by HR). The losses in each cases are computed and transported to the generator module for training. A discriminator network takes in the luma components of both the SR and the HR images and performs a binary classification which trains the generator for photo-realism. Black arrows mark forward computation paths, while red arrows marks gradient backpropagation paths.

Our major contributions in this paper are:

1. Through empirical experiments based on IoU scores, we identify the root causes of artifacts in SRGAN.
2. We propose an alternate algorithm (EaSRGAN) Edge aware Super Resolution Generative Adverarial Network, one that is based on partial treatment of flat and high frequency regions that make EaSRGAN edge aware. We extract Sobel features from LR, HR and SR images and formulate custom loss functions on these edge maps to incorporate the notion of edge-awareness into the network's training process.
3. We perform stability of training analysis, by which we show that training EaSRGAN is free from GAN training stability issues and is comparitively better than SRGAN.

We describe our method in section [2], wherein we discuss about the architecture and the different loss functions (section [2.2]) that are used in training EaSRGAN and we portray the training process in sections [2.2.1, 2.2.2]. We

use IoU scores for performing visual analysis, which we discuss in sections [2.1, 3.2.1]. Finally, we discuss about the performances of EaSRGAN and compare it with that of SRResNet and SRGAN (section [3, 3.2.2]).

We perform stability analysis (section [3.2.3]) of SRGAN and EaSRGAN (examining the variation of PSNR with the number of iterations), and show that EaSRGAN is much more stable than SRGAN. An example of the visual clarity of EaSRGAN is portrayed in Fig. 1.

2 Materials and Method

2.1 IoU Scores

For an image pair SR and HR, we extract three sets of pixels from the luma channels. The three sets of pixels are:

$$P = \{(x,y): \quad 20 \cdot \log_{10}(255 \ / \ |d(x,y)|) < \alpha, \quad d(x,y) \in (SR - HR)\} \tag{1}$$

where $d(x,y)$ represents the difference between pixel intensities of SR (the super-resolved image produced at the output of EaSRGAN) and HR (the ground truth high resolution image). Now we define two set of pixels E and F as follows:

$$\begin{aligned} E &= \{(x,y) : b(x,y) = 1, b(x,y) \in C(HR)\} \ , \\ F &= \{(x,y) : b(x,y) = 0, b(x,y) \in C(HR)\} \end{aligned} \tag{2}$$

where α is an empirically determined threshold. (Our experiments use $\alpha = 20dB$.) P represents the set of 'erroneous' pixels, those that fall below PSNR threshold α. $C(HR)$ is the binary image obtained after processing the HR image through a Sobel edge detector. $b(x,y)$ represents the values of the Sobel edge detector's output. A value of 1 represents edges, and 0 represents non-edge pixels.

We define IoU scores for flat and edge regions separately.

$$IoU_{Flat} = \frac{|P \cap F|}{|P \cup F|}, \qquad IoU_{Edge} = \frac{|P \cap E|}{|P \cup E|} \tag{3}$$

The IoU_{Flat} scores indicate the relative number of 'Erroneous Flat' pixels (pixels which are flat as well as 'erroneous'), to the total number of pixels which are either flat, or 'erroneous'. IoU_{Edge} scores indicate the relative number of 'Erroneous Edge' pixels. The incorporation of information about 'erroneous' pixels allows us to quantify the amount of hallucination performed.

Our EaSRGAN uses region-specific loss functions (for flat and edge regions, i.e., low- and high-frequency regions, respectively), combined with a specific training process. EaSRGAN lets the generator over-hallucinate only on

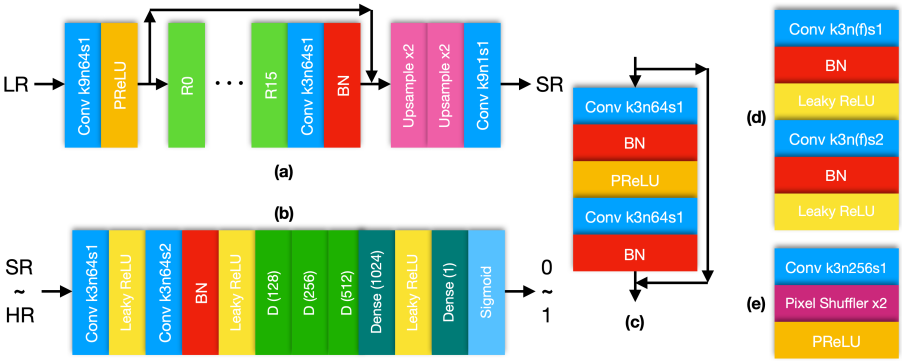


Fig. 3: EaSRGAN architecture. (a) Generator Network (b) Discriminator Network (c) Residual Block (d) Discriminator Module with f filters (e) Upsampler Block. All convolutional layers have the configuration, n filters of shape k x k and stride s.

high-frequency regions and not harm the low-frequency regions. EaSRGAN combines the exactness of SRResNet in low-frequency regions and selectively hallucinate only in the high-frequency regions.

2.2 Architecture and Training

We provide a detailed block diagram of the training procedure in Fig. 2. To enable a fair comparison of our EaSRGAN (Fig. 3) with the state-of-the-art SRGAN [12], we do not make any architectural changes in the basic SRResNet and SRGAN structure itself, by adding or deleting layers.

We do not alter the activation functions either, as altering either has the potential to cause a significant change to the output PSNR scores. This will otherwise prohibit a fair one-on-one comparison of EaSRGAN and SRGAN. This ensures EaSRGAN's improvement to the visual quality and its quantitative measurements do not come from the changes in the network itself. The minor change in EaSRGAN is to apply edge extraction to the luma channel alone (not RGB). This avoids combination issues in having to deal with edges (since EaSRGAN is 'edge-aware') in three R, G and B channels, or using some other colour model. EaSRGAN training or testing uses only the luma (Y) channel. We perform bicubic upscaling on the other two channels (Cb Cr) to match the SR resolution ($4\times$ in our case).

2.2.1 The First Training Phase: Pretraining the Generator

The generator is represented as G_{θ_G}, with parameters $\theta_G = \{W_{1:L}, b_{1:L}\}$: the weights and biases of L layers. In the first phase, we pretrain the generator network with the MSE-based loss function,

$$l_{MSE} = \frac{1}{16R_xR_y}\sum_{x=1}^{4R_x}\sum_{y=1}^{4R_y}(HR_Y(x,y) - G_{\theta_G}(LR_Y)(x,y))^2 \tag{4}$$

Here LR_Y and HR_Y are the luma (Y) channels of the $R_x \times R_y$ LR image, and the HR image, respectively. Hereafter, we denote the output of the generator $G_{\theta_G}(LR_Y)$ as SR_Y, the super-resolved image.

2.2.2 The Second Training Phase: Joint Generator-Discriminator Training, and the Training Schedule

In this phase, we train both the generator and the discriminator networks. Let $C(I)$ be the output of a Sobel edge detector on image I, without non-maximal suppression. The minimum and maximum values of $C(I)$ are normalized in the range $[0, 1]$. To control edge roll-off, we perform additional post-processing to $C(I)$ using a simple exponential transfer function $E(I) = C(I)^\gamma$ to yield our final edge map $E(I)$. (For our experiments, we have empirically chosen $\gamma = 3$.) We compute the edge loss on an HR-SR pair as follows:

$$l_{Edge} = \frac{1}{16R_xR_y}\sum_{x=1}^{4R_x}\sum_{y=1}^{4R_y}(E(HR_Y)(x,y) - E(SR_Y)(x,y))^2 \tag{5}$$

This loss function brings edge-awareness to the generator network. The discriminator is represented as D_{θ_D}, with parameters $\theta_D = \{W_{1:M}, b_{1:M}\}$: the weights and biases of M layers. We define an edge-based adversarial loss function that makes the EaSRGAN discriminator, 'edge-aware' (with N training samples):

$$l_{Eadv} = \sum_{n=1}^{N} -\log D_{\theta_D}(E(G_{\theta_G}(LR_Y))) \tag{6}$$

We consider an edge-aware loss function as a linear combination of the edge- and the adversarial loss, with the linear combination coefficient 10^{-3} the same as that for SRGAN. The final loss is as follows:

$$l_{Gen} = l_{Edge} + 10^{-3} l_{Eadv} \tag{7}$$

For smooth (non-edge) regions, we define a flat loss function:

$$\begin{aligned} l_{Flat} = &\frac{1}{16R_xR_y}\sum_{x=1}^{4R_x}\sum_{y=1}^{4R_y}((1 - E(HR_Y)(x,y))HR_Y(x,y) \\ &- (1 - E(G_{\theta_G}(LR_Y))(x,y))G_{\theta_G}(LR_Y)(x,y))^2 \end{aligned} \tag{8}$$

The second training phase jointly uses both the generator and the discriminator, alternating updates to the generator and the discriminator.

In one iteration of the second training phase:

1. The discriminator is trained with the min-max objective:

$$\min_{\theta_G} \max_{\theta_D} \quad \mathbb{E}_{HR\sim p_{train}(HR)}[\log D_{\theta_D}(E(HR_Y))] + \mathbb{E}_{LR\sim p_G(LR)}[\log(1 - D_{\theta_D}(E(G_{\theta_G}(LR_Y))))] \tag{9}$$

   $E(HR_Y)$ is trained with target label (1) and $E(G_{\theta_G}(LR_Y))$ is trained with target label (0).
2. The generator is trained with the edge-aware l_{Gen} (Eq. 7).
3. A second generator update is performed with l_{Flat}, (Eq. 8). The flat loss function targets the flat regions and does not allow too much deviation from already obtained 'high PSNR'-like smoothness from the first training phase. Thus it maintains image integrity and suppresses artifacts in the flat regions. The flat loss function allows EaSRGAN to fall back to an SRResNet-like solution for image regions that are too 'risky' to be hallucinated. Section 3 shows that these are the key areas where SRGAN loses PSNR, since it generates artifacts in these flat areas.

We use the same protocol for the training images as in SRGAN. We take a random selection of 350,000 images from ImageNet [19]. HR images are bicubically downscaled by 4× to obtain the LR images. Random crops of 96×96 pixels in a batch of 16 HR images are used per iteration. HR images are scaled to range $[-1, 1]$ while LR images are scaled to range $[0, 1]$. An Adam optimizer with $\beta_1 = 0.9$ is used for all optimization stages. The first training phase (Section 2.2.1) uses a learning rate of 10^{-4}, over 10^6 iterations. For the second phase (Section 2.2.2), we use a learning rate of 10^{-5} in step 1, and 10^{-3} in steps 2 and 3. A total of 10^5 iterations are performed in the second phase. Random flips and random rotations are performed with probability 0.5 to augment the data.

3 Experiments

This section compares the merits and the drawbacks of the three approaches. First we present the super-resolution outputs of the three algorithms and then use IoU scores (to analyze artifacts), and examine PSNR issues. We use the same data sets as SRGAN namely, the three widely used benchmark datasets Set5 [20], Set14 [21] and BSD100, the test set of BSD300 [22]. The code for all experiments are made available at https://github.com/bishshoy/easrgan.

3.1 Visual Analysis

Figs. 4, 7, 5 and 6 compare the super-resolved output of SRResNet, SRGAN and EaSRGAN, with a bicubic interpolation and the original HR version. The upscaling factor is 4× in each dimension (i.e., a 16× increase in area). In Fig. 4, we have the upscaling results of the 'Barbara' image from Set14 [21].

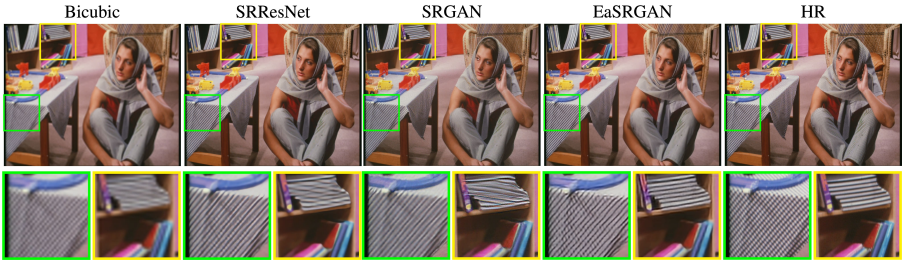


Fig. 4: EaSRGAN reproduces both the books and striped cloth regions of the 'Barbara' image (Set14) successfully. This is due to the awareness of EaSRGAN. It does not suffer from the smoothing in SRResNet, or any unncessary artifacts in SRGAN.

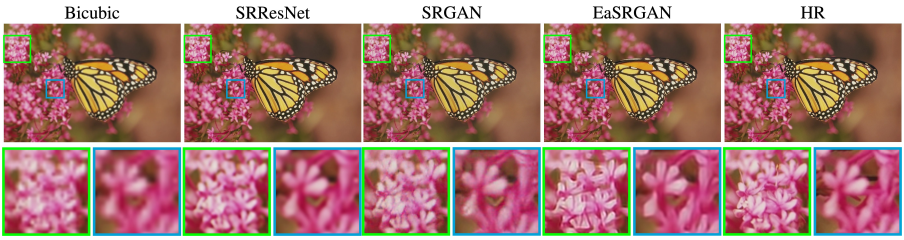


Fig. 5: For the 'Monarch' image (Set14), EaSRGAN outperforms SRResNet and SRGAN in difficult regions such as the pink flowers in the green and the blue box.

The 'edge-awareness' of EaSRGAN's discriminator enables the proper recovery of the books in the bookshelf (the yellow box). The same also reflects in EaSRGAN's reconstruction of the cloth stripes in both directions (the green box). While SRResNet produces a blurred output, SRGAN is unable to pick the stripes in both directions, or the structure of the books: over-hallucination produces too many artifacts (structures and colors not in the original image).

Fig. 7 shows the corresponding results from the 'PPT3' image of Set14. EaSRGAN produces much more readable text than either SRResNet or SRGAN, as emphasized in the red and green boxes.

Fig. 5 shows results for the 'Monarch' image of Set14. The butterfly itself is well-reconstructed by all three algorithms. Let us concentrate on the regions with the pink flowers. In the green boxed region, SRGAN produces severe artifacts, while the SRResNet output is blurry. EaSRGAN maintains the overall sharpness of this region without any visible artifacts. In the blue boxed region, the EaSRGAN output is sharper than the actual HR image itself. We consider this to be a failure case of EaSRGAN, as it does not learn the 'bokeh' effect of the lens with which the image was taken. The edge losses drive EaSRGAN to treat the 'bokeh' effect as downscaled resolution loss, and hence it upscales it with the target of producing sharp edges.

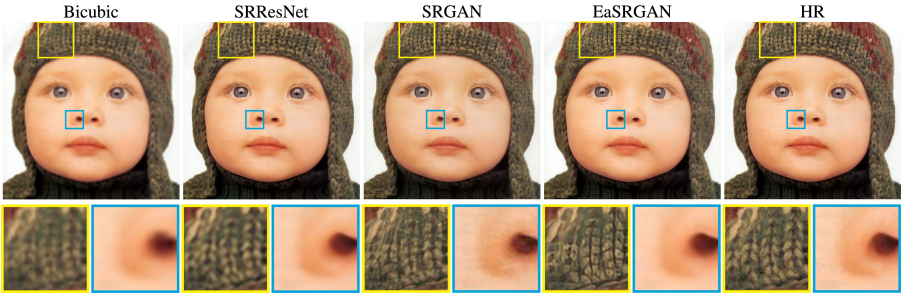


Fig. 6: For the 'Baby' image (Set14), SRResNet performs poorly in regions such as the yellow box, since it has lots of edges. SRGAN on the other hand, hallucinates too much and creates a lot of artifacts. EaSRGAN reconstructs the space in between the stitching of the woolen hoodie. For the blue boxed region, SRGAN produces artifacts, lowering the PSNR. The flat loss function allows EaSRGAN not to let the PSNR fall too much.

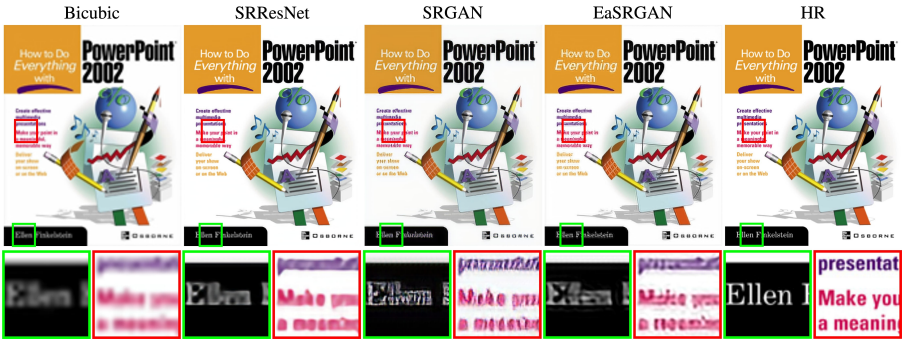
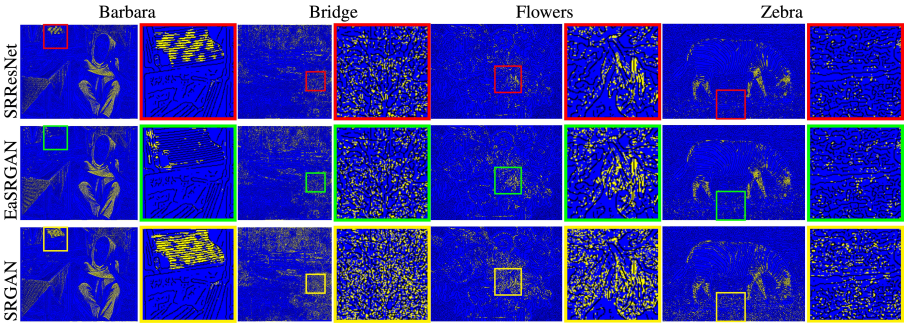
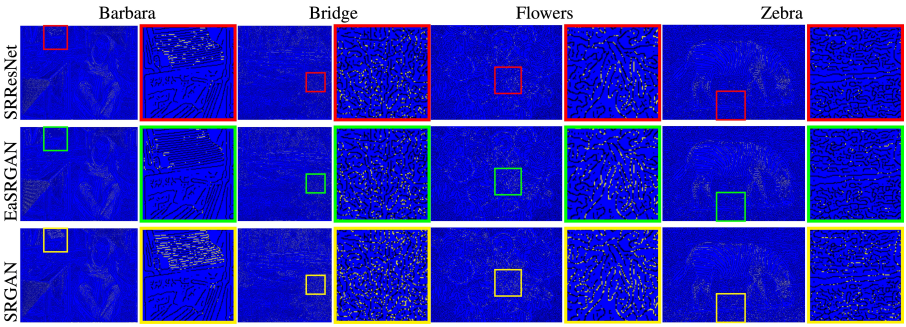


Fig. 7: Results for the 'PPT3' image (Set14). Readers are requested to zoom out and view the above image from a far away distance. SRResNet replaces every letter with a smeared blob. The spacing between the letters are also not maintained, and the letters seem to 'overlap' onto each other. The word Ellen has a water painting effect in SRResNet. SRGAN over-hallucinates and creates a lot of artifacts. The letters are split and the structural integrity of each letter is impaired. In EaSRGAN, the spacing between the letters is maintained which is possible due to its edge-awareness. The second 'e' in the word 'Ellen' (green box) is unclear in both SRResNet and SRGAN. In SRResNet, it is smeared while in SRGAN, the letter is split into multiple components. In EaSRGAN, the second letter 'e' of the word Ellen has clear structural integrity. It is both separated from the adjacent letters and the empty space inside the letter 'e' is also clearly visible. The same goes for the letters 'll' of the word 'Ellen'.

In Fig. 6 EaSRGAN produces sharp results for the 'Baby' image of Set5 [20] (the yellow box). Consider the blue box: the HR image has some very minute non-repeating undulations (hence not 'texture'). This high frequency noise



(a) Visualization of artifacts produced in flat regions which is quantified by the IoU_{flat} measure.



(b) Visualization of artifacts produced on edge pixels, which is quantified by the IoU_{edge} measure.

Fig. 8: Difference images (SR and HR) for 4 representative images of Set14 (Barbara, Bridge, Flowers and Zebra). Yellow pixels represent erroneous pixels. Smaller number of yellow pixels in a given area is better as it indicates a more accurate representation. Of the two photorealistic SR algorithms, EaSRGAN outperforms SRGAN (a smaller number of yellow pixels, without the unnecessary smoothing of SRResNet.) Readers are requested to zoom in while viewing these images. Table 1 shows quantitative IoU results.

gets completely aliased in the downsampled LR image. (No traces of the same are visible in the bicubic image.) SRGAN over-hallucinates and produces more noise than that is present in the original HR image. Due to its flat loss function, EaSRGAN falls back to a more conservative approach in order to produce the highest PSNR version, which is very close to that of SRResNet. The yellow box shows the controlled hallucination property of EaSRGAN, in regions that contain many edges. The blue box shows the conservative property of EaSRGAN in regions where hallucination would lead to artifacts.

Table 1: IoU_{Flat} and IoU_{Edge} scores for all Set14 images. Lower is better. Bold values indicate lower IoU scores among photorealistic SR algorithms.

	IoU_{Flat} Scores			IoU_{Edge} Scores		
	(MSE)	(Photorealistic)		(MSE)	(Photorealistic)	
Image	SRResNet	SRGAN	EaSRGAN	SRResNet	SRGAN	EaSRGAN
Baboon	0.1398	0.1824	**0.1594**	0.0465	0.0616	**0.0534**
Barbara	0.0685	0.0798	**0.0738**	0.0182	0.0224	**0.0212**
Bridge	0.0686	0.1336	**0.0766**	0.0234	0.0414	**0.0263**
Coastguard	0.0511	0.1232	**0.0613**	0.0130	0.0341	**0.0147**
Comic	0.0970	0.1486	**0.1287**	0.0384	0.0547	**0.0515**
Face	0.0015	0.0059	**0.0020**	0.0004	0.0018	**0.0005**
Flowers	0.0251	0.0563	**0.0361**	0.0115	0.0210	**0.0157**
Foreman	0.0130	**0.0200**	0.0214	0.0069	0.0099	**0.0073**
Lena	0.0080	0.0181	**0.0164**	0.0022	0.0062	**0.0052**
Man	0.0364	0.0625	**0.0470**	0.0134	0.0211	**0.0170**
Monarch	0.0066	**0.0128**	0.0142	0.0052	**0.0077**	0.0085
Pepper	0.0064	0.0092	**0.0087**	0.0029	0.0042	**0.0035**
PPT3	0.0239	0.0340	**0.0330**	0.0091	0.0148	**0.0133**
Zebra	0.0340	0.0647	**0.0394**	0.0146	0.0226	**0.0168**

Table 2: PSNR Table for various SR algorithms at 4× upscaling factors. The highest measures are marked in bold. EaSRGAN outperforms other photorealistic algorithms while maintaining an average PSNR close to the best MSE-based SR methods. For reference, the methods mentioned here are SRCNN [1], ESRT [23], SRFBN [24], MSFIN+ [25], SRResNet [12], NLSN [26], ENet-E [15], SRGAN [12], ENetPAT [15] and IEGAN [18]

			MSE-based SR							**Photorealistic SR**			
	NN	Bicubic	SRCNN	ESRT	SRFBN	MS-FIN+	SR-ResNet	NLSN	ENet-E	SRGAN	ENet-PAT	IEGAN	EaSRGAN
Set5	26.26	28.43	30.07	32.19	32.39	32.39	32.05	32.59	31.74	29.40	28.56	-	30.50
Set14	24.64	25.99	27.18	28.69	28.77	28.66	28.49	28.87	28.42	26.02	25.77	25.03	27.47
B100	25.02	25.94	26.68	27.69	27.68	27.61	27.58	27.78	27.50	25.16	24.93	-	26.82

Table 3: PI/PSNR scores on the PIRM dataset [27] for various super-resolution algorithms.

	SRResNet	SRGAN	EaSRGAN
Set5	5.89/32.05	3.36/29.40	5.53/30.50
Set14	5.21/28.49	2.88/26.02	4.79/27.47
PIRM	2.09/28.33	5.18/25.60	4.39/26.92

3.2 Results

3.2.1 IoU scores

In this section, we analyze the presence of artifacts in the SR outputs of the three algorithms by using the formulation of IoU scores discussed in Sec. 2.1. Table 1 shows the IoU scores for all 14 images of the Set14 dataset. Fig. 8 shows a visual depiction of the same, for four representative images of Set14. We request the reader to zoom in to the boxed portions. SRGAN produces the maximum number of erroneous pixels. EaSRGAN performs mostly on par with

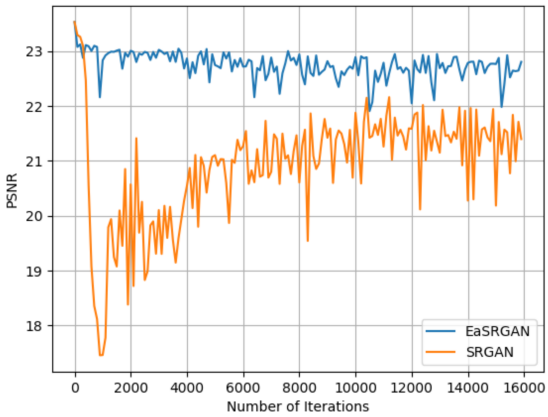


Fig. 9: Stability of EaSRGAN vs. SRGAN: a PSNR plot with the number of iterations shows a much lower standard deviation for EaSRGAN (0.2572, the blue curve) as compared to SRGAN (0.5673, the brown curve). The two start from the same point, but SRGAN's hallucination drops below 18dB. The more stable EaSRGAN maintains a high PSNR level with very little variation.

SRResNet, while maintaining being 'edge-aware', and not allowing unnecessary smoothness. These erroneous areas contribute to the presence of artifacts since the error threshold is chosen to be a very low 20 dB. Further, these areas contribute to the overall drop of PSNR. Among photorealistic SR algorithms, EaSRGAN outperforms SRGAN.

3.2.2 PSNR and PI

The PSNR values obtained by SRResNet, SRGAN, EaSRGAN and various other SR algorithms are presented in the Table 2. EaSRGAN falls in the category of photorealistic SR algorithms as it does not aim to optimize MSE. It outperforms other photorealistic algorithms like SRGAN[[12]] and [EnhanceNet [15]] (the ENet-E and ENet-PAT versions) by a significant margin, attributed to the artifact suppression obtained as a result of edge-aware loss functions and the flat loss function. The PSNR values are obtained on the SR images of each algorithm. The SR images are first converted to YCbCr and only the Y channel is kept, while the other two channels are discarded. We also mention the Perceptual Index (PI) scores on the PIRM dataset [27] for various algorithms in Table 3.

3.2.3 Stability Analysis

We analyze the stability of EaSRGAN, and compare it with that of SRGAN. As a representative example, Fig. 9 measures the rate of PSNR change every 100 iterations, for the generated super-resolved image of 'Comic' from Set14. We notice that both plots start from the same point, which is the PSNR

obtained by SRResNet. SRGAN's implementation of hallucinated details overwhelmingly reduces the PSNR and after a few hundred iterations, the PSNR drops below 18 dB. EaSRGAN maintains a very high PSNR throughout the training process. SRGAN recovers in the end, but settles for an overall low final PSNR as compared to EaSRGAN. To quantize the stability of the overall training process, we compute the standard deviation of the two curves. This helps measure the relative stability of the two networks. We see a standard deviation of 0.2572 in the PSNR curve obtained by EaSRGAN, while the same is 0.5673 for SRGAN. SRGAN over-hallucinates, and creates artifacts that reduce PSNR and makes it unstable. EaSRGAN does not suffer from this problem, as it restricts hallucination in the desired areas.

4 Conclusion

We established a taxonomy of modern SR algorithms, namely in the context of photorealism. We analyze the strength, merits and demerits of SRGAN, a well established photorealistic algorithm. On the basis of the observed caveats of SRGAN and the strength of SRResNet, we propose EaSRGAN that aims to bridge the gap by taking a different route at how artifacts are handled by the GAN. The result is perceptually superior to both SRGAN and SRResNet. We establish the perceptual superiority of EaSRGAN in handling artifacts as a photorealistic algorithm by introducing IoU scores, separate for different sections of an image. We show that EaSRGAN technically performs better than SRGAN in the context of photorealistic algorithm, by a subjective analysis. This points to the conclusion that we have been able to lower the quantity of observed artifacts, that is abundantly found in SRGAN.

5 Compliance with Ethical Standards

- Conflict of Interest: The authors declare that they have no conflict of interest.
- Ethical approval: This article does not contain any studies with human participants or animals performed by any of the authors.

References

[1] Dong C, Loy CC, He K, Tang X. Learning a Deep Convolutional Network for Image Super-Resolution. In: Proc. European Conference on Computer Vision (ECCV); 2014. p. 184 – 199.

[2] Bruna J, Sprechmann P, LeCun Y. Super-Resolution with Deep Convolutional Sufficient Statistics. arXiv preprint arXiv:151105666. 2015;.

[3] Dong C, Loy CC, Tang X. Accelerating the Super-Resolution Convolutional Neural Network. In: Proc. European Conference on Computer Vision (ECCV); 2016. p. 391 – 407.

[4] Johnson J, Alahi A, Fei-Fei L. Perceptual Losses for Real-Time Style Transfer and Super-Resolution. In: Proc. European Conference on Computer Vision (ECCV); 2016. p. 694 – 711.

[5] Kim J, Kwon Lee J, Mu Lee K. Accurate Image Super-Resolution using Very Deep Convolutional Networks. In: Proc. IEEE International Conference on Computer Vision and Pattern Recognition (CVPR); 2016. p. 1646 – 1654.

[6] Kim J, Kwon Lee J, Mu Lee K. Deeply-Recursive Convolutional Network for Image Super-Resolution. In: Proc. IEEE International Conference on Computer Vision and Pattern Recognition (CVPR); 2016. p. 1637 – 1645.

[7] Shi W, Caballero J, Huszár F, Totz J, Aitken AP, Bishop R, et al. Real-time Single Image and Video Super-Resolution using an Efficient Sub-pixel Convolutional Neural Network. In: Proc. IEEE International Conference on Computer Vision and Pattern Recognition (CVPR); 2016. p. 1874 – 1883.

[8] Shi Y, Wang K, Xu L, Lin L. Local-and holistic-structure preserving image super resolution via deep joint component learning. In: Multimedia and Expo (ICME), 2016 IEEE International Conference on; 2016. p. 1–6.

[9] Yu X, Porikli F. Ultra-resolving Face Images by Discriminative Generative Networks. In: Proc. European Conference on Computer Vision (ECCV); 2016. p. 318 – 333.

[10] Adil M, Mamoon S, Zakir A, Manzoor MA, Lian Z. Multi scale-adaptive super-resolution person re-identification using GAN. Ieee Access. 2020;8:177351–177362.

[11] Li X, Du Z, Huang Y, Tan Z. A deep translation (GAN) based change detection network for optical and SAR remote sensing images. ISPRS Journal of Photogrammetry and Remote Sensing. 2021;179:14–34.

[12] Ledig C, Theis L, Huszár F, Caballero J, Cunningham A, Acosta A, 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. p. 4681–4690.

[13] Zhang Y, Li K, Li K, Wang L, Zhong B, Fu Y. Image super-resolution using very deep residual channel attention networks. In: Proc. European Conference on Computer Vision (ECCV); 2018. p. 286 – 301.

[14] Wang Y, Perazzi F, McWilliams B, Sorkine-Hornung A, Sorkine-Hornung O, Schroers C. A Fully Progressive Approach to Single-Image Super-Resolution. arXiv preprint arXiv:180402900. 2018;.

[15] Sajjadi MS, Schölkopf B, Hirsch M. Enhancenet: Single Image Super-Resolution through Automated Texture Synthesis. In: Proc. IEEE International Conference on Computer Vision (ICCV); 2017. p. 4501 – 4510.

[16] Wang X, Yu K, Wu S, Gu J, Liu Y, Dong C, et al. ESRGAN: Enhanced Super-Resolution Generative Adversarial Networks. In: Prof. European Conference on Computer Vision (ECCV) Workshops. vol. V; 2018. p. 63 – 79.

[17] Vasu S, Madam NT, Rajagopalan AN. Analyzing Perception-Distortion Tradeoff using Enhanced Perceptual Super-resolution Network. In: Prof. European Conference on Computer Vision (ECCV) Workshops. vol. V; 2018. p. 114 – 131.

[18] Ghosh SS, Hua Y, Mukherjee SS, Robertson N. IEGAN: Multi-purpose Perceptual Quality Image Enhancement Using Generative Adversarial Network. In: Proc. IEEE Winter Conference on Applications of Computer Vision (WACV); 2019. p. 11 – 20.

[19] Russakovsky O, Deng J, Su H, Krause J, Satheesh S, Ma S, et al. ImageNet Large Scale Visual Recognition Challenge. International Journal of Computer Vision. 2015;115(3):211 – 252.

[20] Bevilacqua M, Roumy A, Guillemot C, Alberi-Morel ML. Low-Complexity Single-Image Super-Resolution based on Nonnegative Neighbor Embedding. In: Proc. British Machine Vision Conference (BMVC); 2012. p. 135.1 – 135.10.

[21] Zeyde R, Elad M, Protter M. On Single Image Scale-up using Sparse Representations. In: Proc. International Conference on Curves and Surfaces; 2010. p. 711 – 730.

[22] Martin D, Fowlkes C, Tal D, Malik J. A Database of Human Segmented Natural Images and its Application to Evaluating Segmentation Algorithms and Measuring Ecological Statistics. In: Proc. IEEE International Conference on Computer Vision (ICCV); 2001. p. 416 – 423.

[23] Lu Z, Liu H, Li J, Zhang L. Efficient transformer for single image super-resolution. arXiv preprint arXiv:210811084. 2021;.

[24] Li Z, Yang J, Liu Z, Yang X, Jeon G, Wu W. Feedback network for image super-resolution. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition; 2019. p. 3867–3876.

[25] Wang Z, Gao G, Li J, Yu Y, Lu H. Lightweight image super-resolution with multi-scale feature interaction network. In: 2021 IEEE International

Conference on Multimedia and Expo (ICME). IEEE; 2021. p. 1–6.

[26] Mei Y, Fan Y, Zhou Y. Image super-resolution with non-local sparse attention. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition; 2021. p. 3517–3526.

[27] Blau Y, Mechrez R, Timofte R, Michaeli T, Zelnik-Manor L. The 2018 PIRM challenge on perceptual image super-resolution. In: Proceedings of the European Conference on Computer Vision (ECCV) Workshops; 2018. p. 0–0.