Generative Adversarial Networks for Super Resolution Imaging

Group 11

Alexander Charles Huang
MPDSC
Chalmers
Gothenburg, Sweden
huanga@student.chalmers.se

Elias Löfgren
MPSYS
Chalmers
Gothenburg, Sweden
elilof@student.chalmers.se

***Abstract*—While the application-space of super-resolution imaging prevails, the need for further exploration and research becomes seemingly more widespread within the field. This paper highlights three conceptually progressive generative adversarial network models constructed through building upon previous versions, taking inspiration from revolutionary proposed methodologies and state-of-the-art models. The study suggests that the modifications brought by the ESRGAN model built on top of the original SRGAN allow for consistently better results. Lastly an approach to using inception-residual blocks in conjunction with the ESRGAN model which could be further extended and explored is presented in this report.**

I. Introduction

Super-resolution (SR) is a class of problems that aims to generate high resolution images from low resolution images. A high resolution image generally convey more detail about an original scene in comparison to a low resolution image. Hence the interest of solving this type of problem is vast within various areas related to image processing. Applications of SR where high resolution images are of importance are e.g. medical image processing for diagnosis, satellite imaging for identifying target fields, multimedia and video enhancement [1].

As there exist an abundant amount of techniques to enhance the resolution of images, deep machine learning methods approaches have notably emerged within the solution space [2]. One type of network that have emerged in particular is the generative adversarial networks (GANs), which is a framework consisting of two neural network architectures essentially competing with each other in order to improve and perfect the generated content.

II. Related work

A. Generative Adversarial Networks (GANs)

The concept of GANs was originally proposed in the paper *Generative adversarial nets*, by I.J. Goodfellow et al [3]. The paper introduced a seemingly revolutionary framework for generating artificial data with neural networks. The overarching idea is to simultaneously train two models, namely a generator and a discriminator, where the generator is trained to generate data as closely distributed to the ground truth data as possible while the discriminator is trained to estimate the probability that a sample came from the generator contra the ground truth data. Essentially the generators objective is to generate data that is so indistinguishable from the ground truth data such that it maximizes the discriminators probability of miss classifying samples. In contrast the discriminators objective is to minimize the probability of miss classifying samples. Thus the two models are as described by the authors, playing a "minimax two-player game".

In the original paper the following notions are introduced

- p_g - Generator's distribution over data x
- z - Noise or generator input data
- $G(z;\theta_g)$ - Differentiable function representing the generator with parameters θ_g. The output lies in p_g.
- $D(x;\theta_d)$ - Differentiable function representing the discriminator with parameters θ_d. The output is a scalar that represents the probability that x came from the real data rather than p_g

The discriminator (D) is trained to maximize the probability of correctly assigning labels to samples of training data and samples from the generator (G), which means to maximize the following expression

$$\frac{1}{m}\sum_{i=1}^{m}\log D(x^{(i)}) + \log\left(1 - D(G(z^{(i)}))\right)$$

G is trained concurrently to minimize the corresponding expression

$$\frac{1}{m}\sum_{i=1}^{m}\log\left(1 - D(G(z^{(i)}))\right)$$

Interestingly these two expressions are directly derived from the general optimization problem formulation which is presented in the original paper.

$$\min_G \max_D V(D,G)$$
$$= E_{x\sim p_{data}}[\log D(x)] + E_{z\sim p_g(z)}[\log\left(1 - D(G(z))\right)]$$

B. Super Resolution GANs

Before GANs were invented, one approach for SR was to train a single deep convolutional neural network model and obtain a loss on the pixel-wise mean squared error (MSE) between the generated and the ground truth image [4]. Soon after the invention of GANs, the introduction of a discriminator allowed for an adversarial loss in conjunction with the traditional pixelwise loss [5]. However, one of the most compelling and well performing architectures was proposed by C. Ledig et al [2]. This model is called SRGAN (Super Resolution GAN), which aims to utilize a pretrained VGG-network to extract features and calculate the MSE loss between the generated and the ground truth image. Thus the generator loss function is a content loss based on VGG-extracted features summed with the traditional adversarial loss, resulting in a feature based perceptual loss function. In detail, the loss is a weighted sum of the two loss functions captured in the following expression

$$L_G = c_{VGG-content} \cdot L_{VGG-content} + c_{adversarial} \cdot L_G^{adv}$$

This allows G to consider finer texture details when performing SR at large up-scaling factors, which before SRGAN was a central problem. To rephrase it, G motivates it's content loss on the perceptual similarity instead of simple similarity in pixel space.

As can be seen in figure 1, the SRGAN's G network utilizes B number of uniformly structured residual blocks with one skip connection in each. A residual block consists of 2 convolutional layers, each including batch normalization were only the first layer is followed by a parametric rectified linear unit (PReLU) activation function. The last operator of the block is the elementwise sum which exist due to the skip connection in the block. Furthermore a skip connection is set to go from the output of the first convolutional layer to the last convolutional layer before the upscaling blocks. Subsequently, there are two upscaling blocks each consisting of a convolutional block, pixel shuffle operation with an upscaling factor of 2 followed by batch normalization.

The D network is a straightforward deep convolutional classifier with progressively larger number of output channels in each convolutional layer. The classifier layers consist of a simple feed forward network with a dense linear layer, leaky ReLU followed by an output layer with a single node. The output is forwarded through a sigmoid activation function to be able to obtain a binary prediction value when doing forward propagation.

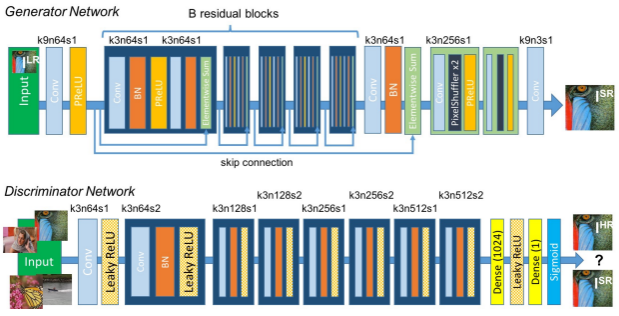

Fig. 1. SRGAN model architectures. Source: Adapted from [2]

C. Enhanced super resolution gan

Further research and exploration prompted an extension of the original SRGAN. One of which produced remarkable results and developed into what is considered to be state-of-the-art within super-resolution. It was labeled as enhanced super resolution GAN (ESRGAN) and proposed by X. Wang et al [6]. The ESRGAN network utilizes the same basic principles as SRGAN but with some modifications and improvements intended to increase performance and decrease computational cost.

One of the changes incorporated in the ESRGAN architecture is the residual in residual dense block seen in fig 2, where the batch normalization (BN) layers are excluded and more skip connections are included. As motivated in [6], the exclusion of BN layers is done to avoid visually unpleasing image artifacts as observed empirically in the study. Furthermore, excluding BN layers has proven to increase performance and reduce computational complexity in SR [7]. Lastly, the additional dense skip connections are included to improve the networks capacity by enabling deeper network capabilities.

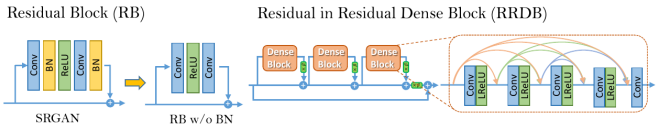

Fig. 2. Residual block (SRGAN) vs. Residual in Residual Dense Block (ESRGAN) taken from [6]

As for the discriminator, the ESRGAN model uses a relativistic discriminator. Instead of letting the discriminator output a probability that a sample is real or generated, the relativistic average discriminator (denoted as D_{Ra}) tries to predict the probability that a real sample x_r is relatively more realistic than a fake sample x_f. Thus the discriminator and generator (adversarial) loss becomes the following respectively

$$L_D = -E[log(D_{ra}(x_r, x_f))] - E[\log(1 - D_{Ra}(x_f, x_r))]$$

$$L_G^{adv} = -E[\log(1 - D_{Ra}(x_r, x_f))] - E[log(D_{ra}(x_f, x_r))]$$

where $E[.]$ is the operator averaging over a mini batch and $x_f = G(x_i)$.

Like in the SRGAN content loss, the ESRGAN content loss for the generator is based on the comparison of features between generated and real image outputted by the VGG19 network. The minor difference is that the ESRGAN utilizes the layers such that the output layer is before an activation layer rather than after an activation layer. The authors of ESRGAN observed that the features after activation became increasingly more vacant as the layers became deeper. Thus extracting features before activation yielded in more information being harnessed. Moreover the total loss of the generator also considers the pixelwise error as an L1-norm between the generated and target image. Thus the generator loss is expressed as

$$L_G = c_{VGG} \cdot L_{VGG} + c_{adv} \cdot L_G^{adv} + c_{pixel} \cdot L_{pixel}$$

D. Inception-Residual Block

Building on the idea of residual blocks enabling deeper networks without vanishing gradient problems and inception modules enabling more efficient computations and utilization of multiple sized kernels along a stacked module, [8] proved to take advantage of the best of both worlds for thermal image denoising. The paper states that architectural inspiration was drawn from Inception-ResNet, which proposes a method to integrate a residual skip connection internally in an inception module as seen in fig 3 [9].

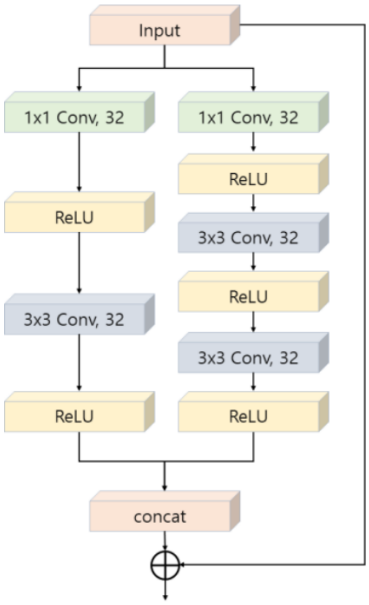

Fig. 3. Inception residual block. Source: Adapted from [8]

III. Method

The models explored in this study were constructed on the basis of SRGAN with some of them including numerous improvement modifications derived from ESRGAN. In total, three different versions of models were conducted and studied analytically. We denote these as V_1, V_2, V_3. The purpose of V_1 is to replicate the original SRGAN model. V_2 is constructed as an extension of V_1 and includes various improvements from ESRGAN. Additionally V_3 is a product of exploration and further investigation. Note that the discriminator architecture is identical to the one in [2] in all three versions and is not presented in detail in this section. All models were trained on COCO dataset, which is mainly used for large scale object detection and segmentation. It is also worth noting that all generators were pretrained with pixelwise loss before the adversarial training.

A. Version 1 (V_1)

As stated in the previous paragraph, V_1 is a replication of SRGAN. G is constructed as described in section II-B with 18 residual blocks. The content loss is computed using the first 18 layers of the VGG19 network. It is rather unclear if the authors of SRGAN included a pixelwise loss, but based on observations of subpar performance without it, the G loss also considers a mean squared error between pixel values of generated and ground truth image. To be specific, the generator loss is defined by

$$L_G = 0.001 \cdot L_{pixel} + 0.006 \cdot L_{VGG} + 0.001 \cdot L_G^{adv}$$

ADAM optimizer is utilized with a learning rate of 0.0001 during training for both G and D.

B. Version 2 (V_2)

In V_2 all BN layers are excluded and instead of PReLU activation layers, Leaky ReLU is incorporated to minimize the number of parameters. Pixelshuffle layers are replaced by simple upsample layers with the mode set to ”nearest” to upscale the image. The content loss is computed using the first 35 layers of the VGG19 network and the weights in G loss function are tuned such that the generator loss is defined by

$$L_G = 0.01 \cdot L_{pixel} + 1 \cdot L_{VGG} + 0.005 \cdot L_G^{adv}$$

Note that L_G^{adv} is computed using a relativistic average discriminator D_{Ra}, which is also one of the improvements made on top of V_1. Hence the improved loss for G and D is described in section II-C.

Despite employing several improvements proposed in ESRGAN, the residual in residual dense block was not considered in V_2 due to time limitations in the project.

C. Version 3 (V_3)

Going further, V_3 replaces the regular residual block with the inception-residual block described in section II-D. Number of blocks is increased from 18 to 24.

IV. Result

In this section the results of our progressively modified versions of models are composed and assessed by peak signal-to-noise ratio (PSNR) and structural similarity index measure (SSIM). PSNR and SSIM are methods used to quantify the reconstruction quality of images. Ultimately higher is better.

It is evident in fig 4 from Appendix that the advancement from V_1 to V_2 resulted in a major performance increase. In contrast, going from V_2 to V_3 resulted in less convincing

performance increase (in terms of PSNR and SSIM). Majority of images slightly favored V_2, albeit the difference is almost negligible. As can be seen in the V_1 images, the brightness is generally higher and BN artifacts are apparent in some instances (e.g. the dandelion on the V_1 bee image). As opposed to V_1, both V_2 and V_3 performed good consistently with sharper edges and overall better color balance. All things considered, it is obvious that no model generated an image that was worse than the low resolution (LR) image.

V. Conclusions

It is once again empirically proven that the improvements from ESRGAN yield in a substantial performance increase. Although the performance difference between V_2 and V_3 is miniscule, one has to consider that the modifications between these versions are also relatively small in comparison to the modifications between V_1 and V_2. Furthermore there could be other aspects of V_3 which yield in a positive net result in regards to other metrics unexplored in this study, as for instance computational speed, memory consumed etc. It is also worth considering that the introduction of inception-residual blocks could bring added dimensionality and other architectural flexibilities that were previously not possible with regular residual or RRDB blocks. This topic could be further explored and analyzed in future studies and is greatly encouraged. Additionally further improvements which would funnel into V_4 could be to fully commit to each of the improvements (excluding RRDB) proposed in the ESRGAN paper. Some of which are MINC-loss (usage of VGG19 which is trained on material recognition), more advanced training techniques (sophisticated learning rate decay by halving it at specified epoch numbers) and network interpolation which can be read more about in the paper.

References

[1] A. Singh and J. Singh, "Super resolution applications in modern digital image processing," *International Journal of Computer Applications*, vol. 150, pp. 6–8, 09 2016.

[2] C. Ledig, L. Theis, F. Huszar, J. Caballero, A. Cunningham, A. Acosta, A. Aitken, A. Tejani, J. Totz, Z. Wang, and W. Shi, "Photo-realistic single image super-resolution using a generative adversarial network," 2017.

[3] I. J. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, "Generative adversarial networks," 2014.

[4] C. Dong, C. C. Loy, K. He, and X. Tang, "Image super-resolution using deep convolutional networks," 2015.

[5] Q. Yan, "Dcgans for image super-resolution , denoising and debluring," 2017.

[6] X. Wang, K. Yu, S. Wu, J. Gu, Y. Liu, C. Dong, C. C. Loy, Y. Qiao, and X. Tang, "Esrgan: Enhanced super-resolution generative adversarial networks," 2018.

[7] B. Lim, S. Son, H. Kim, S. Nah, and K. M. Lee, "Enhanced deep residual networks for single image super-resolution," 2017.

[8] S. Hwang, G. Yu, H. T. Nguyen, N. Shahid, D. Sin, J. Kim, and S. Na, "Inception-residual block based neural network for thermal image denoising," 2018.

[9] C. Szegedy, S. Ioffe, V. Vanhoucke, and A. Alemi, "Inception-v4, inception-resnet and the impact of residual connections on learning," 2016.

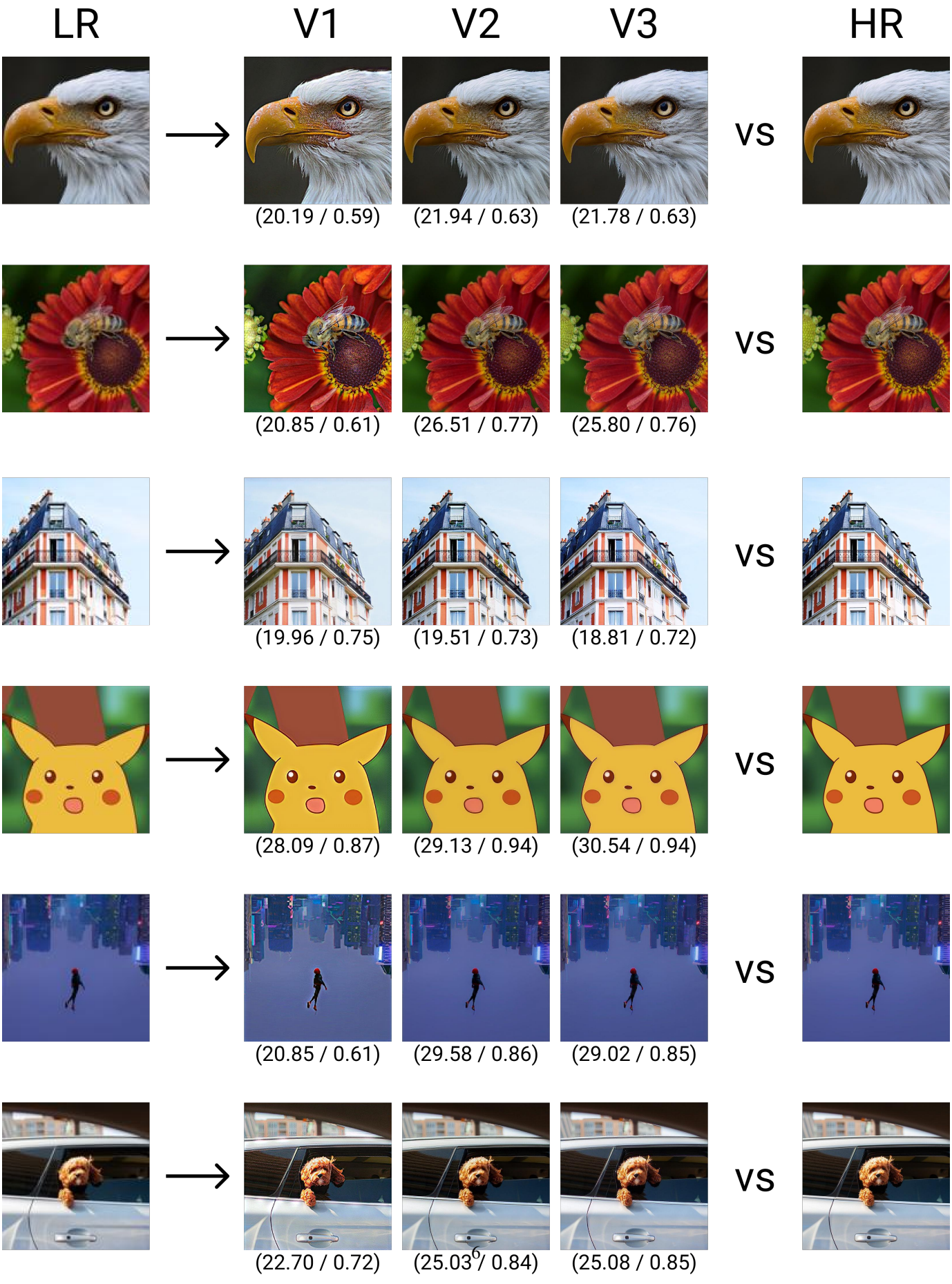

Fig. 4. Results of the generated images (from LR) in comparison to HR. Statistics are given in (PSNR (dB) /SSIM).