INFRARED IMAGE SUPER RESOLUTION WITH DEEP NEURAL NETWORKS

Kyle Vassilo, Tarek Taha

University of Dayton
Dayton, OH
{vassilok1, ttaha1}@udayton.edu

Asif Mehmood

Air Force Research Laboratory
Wright-Patterson AFB, OH
asif.mehmood.1@us.af.mil

ABSTRACT

Recent studies have shown that Deep Learning (DL) algorithms can significantly improve Super Resolution (SR) performance. Single image SR is useful in producing High Resolution (HR) images from their Low Resolution (LR) counterparts. The motivation for SR is the potential to assist algorithms such as object detection, localization, and classification. Insufficient work has been conducted using Generative Adversarial Networks (GANs) for SR on infrared (IR) images despite its promising ability to increase object detection accuracy by extracting more precise features from a given image. This work adopts the idea of a relativistic GAN that utilizes Residual in Residual Dense blocks (RRDBs) for feature extraction, a novel residual image addition, and a Pixel Transposed Convolutional Layer (PixelTCL) for up-sampling. Recent work has validated the use of GANs for Visible Light (VL) images, making them a strong candidate. The inclusion of these components produce more realistic and natural features while also receiving superior metric values.

***Index Terms*—** Deep Learning, Super Resolution, Generative Adversarial Network, Infrared Imaging

1. INTRODUCTION

In recent years, Super Resolution (SR) research has highly depended on Deep Learning (DL) algorithms to learn a set of parameters that map Low Resolution (LR) input images to High Resolution (HR) output images. Some disciplines require images that capture unique frequencies of electromagnetic radiation on the electromagnetic spectrum. For instance, infrared (IR) sensors succeed in detecting infrared radiation by analyzing signals with wavelengths longer than those found in the visible light spectrum. These sensors can highlight objects radiating thermal energy; therefore, allow us to perceive items hidden to the human eye. While IR imaging excels in highlighting thermal energy it lacks in retaining high frequency information, making the SR problem more difficult.

Generative Adversarial Networks (GANs) are one of the novel ideas, designed by Goodfellow in 2014 [2], that have changed the way researchers look at DL. They have proved themselves in the realm of visible light (VL) imagery, by mimicking distributions of image data [3]. We hope that their success with VL imagery provides a promising approach for IR images. Therefore, we will apply a GAN to increase the resolution of IR images. GANs benefit from the adversarial competition between networks [4], where the discriminator acts as a learnable loss function for the overall network. It evaluates the generator's performance, of counterfeiting images, and updates its weights accordingly [4].

The novel contributions of this work are as follows:

- Proposing to inherit components of the ESRGAN network to up-sample IR images while retaining high frequency and IR information [1].
- Proposing to use a Pixel Transposed Convolutional Layer to up-sample the generated images [5]. This technique has yet to be used in a GAN setting.
- Proposing a Residual image addition, where the residual image is calculated to ensure high frequency information propagates through the network.

2. RELATED WORK

GANs have drawn much attention since their discovery in 2014 [2]. Ma et al. proposes a FusionGAN that requires multiple images of the same area. It starts by sending a concatenation of VL and IR images to the generator of a GAN [6]. In most cases, this algorithm is inapplicable for SR procedures because of its dual image requirement. Axel-Christian Guei et al. presents a SRGAN devised to up-sample facial IR images, called DeepSIRF 2.0 [7]. Their network consists of a generator with eight residual blocks and 2 pixel shuffling blocks used for up-sampling [7]. The discriminator consists of seven convolutional layers matched with batch normalization and swish activation functions, ending with a dense layer and a sigmoid function [7]. Wang et al. suggest an Enhanced Super Resolution GAN (ESRGAN), for VL images, that introduces a new Residual in Residual Dense Block (RRDB) [1]. In this article, Wang et al. outline a unique normalization function, known as Spectral Normalization (SN). SN was introduced by Takeru Miyato et al. to normalize layer weights inside the discriminator network using Lipschitz continuity to promote sta-

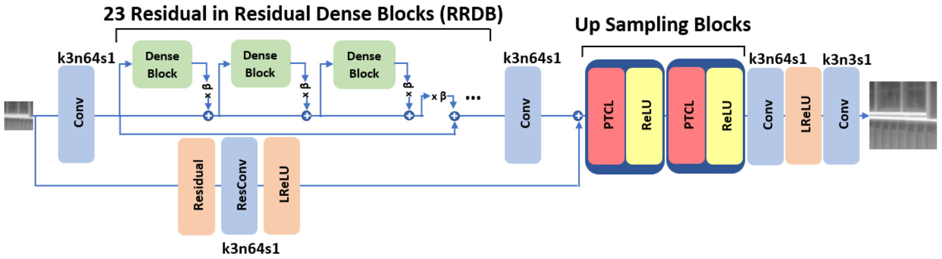


Fig. 1: Generator architecture with corresponding kernel size (k), number of feature output maps (n), and stride (s) for each convolutional layer. A single RRDB is shown, but the generator contains 23 RRDBs that each include 3 dense blocks [1].

bility between the generator and discriminator networks [8]. The ESRGAN also adopts the idea of a Relativistic average GAN (RaGAN) from Alexia Jolicoeur-Martineau [9]. This model's main advantage is its ability to surpass standard SRGAN image quality in less iterations [9]. The ingenuity lies in the loss function of the generator and discriminator, where the probability that the real image is more realistic than the fake image is calculated [9].

3. PROPOSED METHOD

The proposed generator incorporates RRDBs [1], PixelTCLs [5], and a new residual image addition function. The generator network is shown in Figure 1. Examining the first convolutional layer, we can see that the network uses a small kernel size. It was thought that a larger kernel size would help extract initial features from the image, but this turned out to be misleading.

The discriminator network inherits its architecture from the ESRGAN [1], uses 10 convolutional layers paired with SN and a LeakyReLU activation function [8], and ends with 2 dense layers (the first outputting 100 nodes and the last outputting 1 node). The discriminator accepts single channel images and returns a single value without the ending sigmoid function to introduce the relativistic average concept.

3.1. Residual in residual dense block

SRGANs use residual blocks to ensure certain features propagate through the system. A single RRDB can be seen within the generator in Figure 1. Each RRDB contains three residual dense blocks that are multiplied by a scalar value, in our case we chose 0.2 as β, before being added back to the RRDB branch [1]. According to Yungang Zhang et al., removing the batch normalization function within residual blocks can improve its performance by expanding the model size [10]. Furthermore, Xintao Wang et al. explain how multiplying each block by a constant β value can avoid the instability of removing the batch normalization layers [1]. The components of a dense block can be seen in Figure 2. Each convolutional layer within the dense block uses a kernel size of 3, a stride of 1, and outputs 32 feature maps. The input number of feature maps grows as more and more previously calculated layers are concatenated to the input of subsequent layers. The dense connections characterized by this block are inherited from a GAN model introduced by Marc Bosch et al. [11]. These dense connections perform exceptionally well by maintaining a strong gradient flow within the network [11].

3.2. Pixel transposed convolutional layer

Networks that require the functionality of up-sampling data have recently been using techniques such as transposed convolution or sub-pixel convolution [12]. Both techniques have the potential of adding artifacts to the super-resolved image, representing a checkerboard pattern. Figure 3 demonstrates the PixelTCL up-sampling method [5]. The PixelTCL begins by performing a convolution with a kernel size of 3. The resulting tensor is then dilated by inserting rows and columns of zeros to construct a tensor double the size, illustrated by output3 in Figure 3. A second convolution is performed on output1 to produce output2. This tensor is then dilated and shifted to provide output4. The dilated tensors are added together to construct a tensor where every other value is zero. The next convolution exploits a masked kernel where certain values are set to 0. This convolution generates the missing values which are then added to output5 to create the desired output. This technique ensures that each pixel in the resulting image has a strong relationship with its adjacent neighbors; therefore, excluding the checkerboard pattern [5]. The results will prove beneficial as this is a novel technique with respect to SR with GANs.

3.3. Residual image addition

In image processing, a residual image contains the high frequency components of an image by subtracting a sampled image from the original image. In SR, the high frequency elements are what the generated image is missing. To ensure that the high frequency components are not dropped, the residual image is added back to the main branch of the network before up-sampling, similar to Yuewen Sun et al. network [13]. The residual image is created by first down-sampling the input image (plus bicubic interpolation) from a 32×32 sized image to

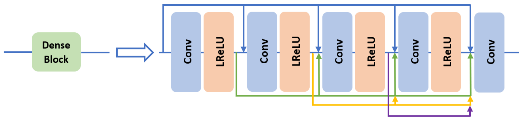


Fig. 2: The dense block within the RRDB [1].

a 24×24 sized image, and re up-sampling it using bicubic interpolation. This image is then subtracted from the original input image to obtain the residual image. The residual image is then fed through a convolutional layer to extract high frequency features, from the residual image, before being added back to the main branch of the network. This approach differentiates itself from Yuewen Sun et al. network by using a different sampling ratio, a shallower skip connection, and adding the results back to the main branch of the network before up-sampling.

3.4. Spectral normalization

The original SRGAN uses batch normalization to normalize the data into values of the same range [4]. Batch Normalization helps a network train faster and go deeper by feeding similar distributions of data to the activation functions that follow. This decreases the amount of training time by creating similar distributions for each layer. SN is a technique used to recalculate the weight of a layer (W) [8]. These normalized weights help stabilize the discriminator of a GAN by dividing each weight value by σ(W):

$$\sigma(W) = \sqrt{\lambda} \tag{1}$$

where λ is the largest eigenvalue of W^TW [8].

3.5. Relativistic average GAN

The original SRGAN employs a discriminator that distinguishes real from fake images. It calculates the probability that the given data is real (returns a 1 if it believes it is an original image from the dataset). The relativistic GAN manipulates both the generator and discriminator loss functions by measuring the distance between the real and fake data [9]. This is done by subtracting the real and the fake data received from the discriminator. Going a step further, the RaGAN calculates the probability that the real data is more realistic than the fake data, just like the relativistic GAN [9], but the average of the opposite label is then subtracted from the given label [9]. The RaGAN is calculated by:

$$\begin{aligned} L_D^{RaGAN} = &-E_{x_r\sim P}[\log(sigmoid(C(x_r) - E_{x_f\sim Q}C(x_f))] \\ &-E_{x_f\sim Q}[\log(1 - sigmoid(C(x_f) - E_{x_r\sim P}C(x_r)))] \end{aligned} \tag{2}$$

$$\begin{aligned} L_G^{RaGAN} = &-E_{x_f\sim Q}[\log(sigmoid(C(x_f) - E_{x_r\sim P}C(x_r))] \\ &-E_{x_r\sim P}[\log(1 - sigmoid(C(x_r) - E_{x_f\sim Q}C(x_f)))] \end{aligned} \tag{3}$$

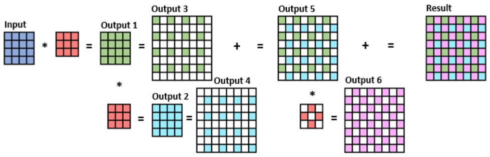


Fig. 3: The PixelTCL used to up-sample image data in the generator [5].

with x_r being the reference image, x_f being the generated image, and $C(\cdot)$ representing the discriminator without its final activation function. The subtraction of the average is demonstrated inside the sigmoid function of equation 2 and equation 3, which illustrates the distance from real to fake data [9].

4. EXPERIMENTS

4.1. Datasets

The IR training images are taken from the FLIR dataset. These images are randomly cropped using MATLAB to produce 20,000 128×128 training images. These images are initially sent through a Gaussian filter, with a standard deviation of $\sigma = 1.3$, down-sampled by a factor of 4, and interpolated using bicubic interpolation [14]. The Gaussian filter is used to represent the image being taken from a camera at a further distance. The images are then concatenated to produce a three-channel image, where each channel contains the same values. This is required to calculate the perceptual loss (VGG loss). Therefore, the generator accepts a 3 channel image and outputs a single channel image.

The IR testing dataset hand picks 1,200 images from the FLIR testing dataset that are randomly cropped to create a working dataset. Following the cropping, images with little to no high frequency content are thrown out, leaving us with a test dataset containing 211 images.

4.2. Implementation details

The network was run on 2 NVIDIA RTX 2080ti GPUs for 50 epochs with a batch size of 16, using Adam with decay rates of 0.9. The learning rate for both the generator and discriminator start at 2×10^{-4} while being reduced by a factor of 2 every 10 epochs. The generator is pre-trained for 2 epochs using Mean Squared Error (MSE). After the two initial epochs, the generator weights are saved and loaded back into the designed model where the Binary Cross Entropy (BCE) function is used. The generator and discriminator loss functions integrate the relativistic idea with BCE by computing the relativistic operations before sending it to the BCE function. The BCE function being used combines a sigmoid function with the BCE loss function to acquire a final discriminator result. The network's generator loss function consists of BCE loss, per-pixel loss, and perceptual loss [1]. The perceptual loss is calculated using a pre-trained VGG19 network and measuring

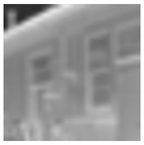
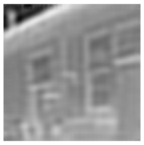
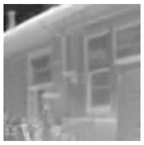
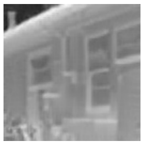
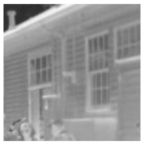
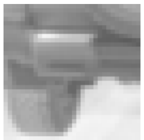
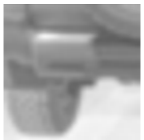
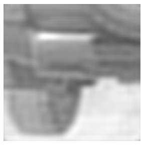
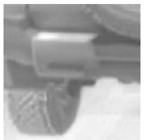
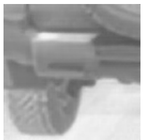
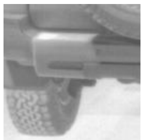
Original Image	Nearest Neighbor	Bicubic	DeepSIRF 2.0 [7]	Proposed Ablation Network	Proposed Network	HR
						
PSNR/SSIM	27.37/0.704	28.71/0.751	28.95/0.747	*30.56/0.801*	**30.91/0.811**	∞/1.00
PIQE	69.70	92.98	77.02	**42.80**	*43.17*	48.04
						
PSNR/SSIM	29.39/0.775	31.82/0.846	32.42/0.835	*33.52/0.877*	**33.88/0.886**	∞/1.00
PIQE	68.82	92.51	77.97	*41.24*	**40.22**	61.99

Fig. 4: Qualitative and Quantitative test results (Best results in **bold**, second best in *italics*).

Table 1: Quantitative comparison between the proposed method and other competing SR algorithms (Best results in **bold**, second best in *italics*).

Up-Sampling Technique	PSNR	SSIM	MSE	PIQE
High Resolution	∞	1.00	0.00	40.47
Nearest Neighbor	25.05	0.600	228.87	65.12
Bicubic	25.96	0.651	188.22	91.39
DeepSIRF 2.0 [7]	*26.24*	*0.661*	*177.24*	80.03
Proposed Ablation Network	26.15	0.660	184.20	**28.57**
Proposed Network	**26.66**	**0.685**	**165.49**	*34.03*

the Least Squares Error (L2) between the real image feature space and the generated image feature space. The per-pixel loss is calculated by measuring L2 distance between the real image and the generated image.

4.3. Results

The results are compared to the DeepSIRF 2.0 network, bicubic interpolation, nearest neighbor interpolation, and an ablation network. The 'Proposed Ablation Network' removes the pre-training procedure, replaces PixelTCL with sub-pixel convolution, and removes the residual image addition in the global residual connection to show how much these techniques add to the overall performance. PIQE is a fairly new metric provided through MATLAB. It is a no-reference quality metric that returns high values for images that are blurred or contain large amounts of noise. A lower score is more acceptable; under 20 being exceptionally well. The average quantitative results are presented in Table 1.

A benchmark testing dataset is desperately needed for IR SR. This benchmark will allow easy comparison between methods, without having to recreate competing methods separately. As of now there is no such IR testing benchmark, so a custom testing dataset was created for comparison. Our proposed method achieved the best quantitative scores in Table 1, for PSNR, SSIM, and MSE. The ablation network scores the best in PIQE, with the proposed network close behind. Looking at Figure 4, it is easy to see that these two methods create the most realistic results. Therefore, we propose to use PIQE as the main metric for IR SR. The ablation network and proposed network results look similar, but at a closer look we can see that the proposed network is cleaner around edges and other high frequency areas. In Table 1, the DeepSIRF 2.0 network scores better than the ablation network in PSNR, SSIM, and MSE. However, the images created by the DeepSIRF 2.0 network illustrate a blurry characteristic that contributes to their poor PIQE scores. Bicubic interpolation also scored well in every category except PIQE. The images generated by bicubic interpolation have lost their high frequency components. Therefore, it receives good PSNR, SSIM, and MSE scores because the low frequency components are very similar to that of the original image, but yields a poor PIQE score due to their blurred appearance. The nearest neighbor interpolation performs exactly how we would expect for PSNR, SSIM, and MSE. These results are fairly low due to the images' discontinuity. Surprisingly, its PIQE score is better than bicubic interpolation and the DeepSIRF 2.0 network. PIQE scores blurred and noisy images poorly, but the blurred images receive a worse score. It is obvious to human inspection that nearest neighbor interpolation is inadequate to up-sample images, so it is easy to throw out despite its good PIQE score. However, when it comes to high level representation methods, PIQE becomes a reliable metric to score the results.

Figure 4 shows that the proposed method generates images of the highest quality, backed by their metric values. Our model preserves most of the textural characteristic, of the LR image, that have not been made entirely obsolete by downsampling. The details in certain images have been recreated, allowing us to distinguish objects in the super-resolved image.

5. CONCLUSION

The generator adopts its residual architecture from the ESRGAN [1] and its up-sampling procedure from Hongyang Gao et al. [12]. These components help the network produce images of higher quality by creating more connections between layers that aid in learning to mimic the data's distribution. The residual image addition and pre-training techniques also add to the generators success. Comparing the proposed network to the ablation network, we can see that these features help the network learn a better representation. Most of the high frequency components from the original image are present in the generated image, even though IR images lack certain high frequency information. Our method is very powerful and can be applied to IR SR to generate highly accurate images that retain IR information.

6. REFERENCES

[1] Xintao Wang, Ke Yu, Shixiang Wu, Jinjin Gu, Yihao Liu, Chao Dong, Yu Qiao, and Chen Change Loy, "Esrgan: Enhanced super-resolution generative adversarial networks," in *The European Conference on Computer Vision (ECCV) Workshops*, September 2018.

[2] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio, "Generative adversarial nets," in *Advances in Neural Information Processing Systems 27*, Z. Ghahramani, M. Welling, C. Cortes, N. D. Lawrence, and K. Q. Weinberger, Eds., pp. 2672–2680. Curran Associates, Inc., 2014.

[3] Md. Zahangir Alom, Tarek M. Taha, Christopher Yakopcic, Stefan Westberg, Mahmudul Hasan, Brian C. Van Esesn, Abdul A. S. Awwal, and Vijayan K. Asari, "The history began from alexnet: A comprehensive survey on deep learning approaches," *CoRR*, vol. abs/1803.01164, 2018.

[4] Christian Ledig, Lucas Theis, Ferenc Huszar, Jose Caballero, Andrew P. Aitken, Alykhan Tejani, Johannes Totz, Zehan Wang, and Wenzhe Shi, "Photo-realistic single image super-resolution using a generative adversarial network," *CoRR*, vol. abs/1609.04802, 2016.

[5] H. Gao, H. Yuan, Z. Wang, and S. Ji, "Pixel transposed convolutional networks," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, pp. 1–1, 2019.

[6] Jiayi Ma, Wei Yu, Pengwei Liang, Chang Li, and Junjun Jiang, "Fusiongan: A generative adversarial network for infrared and visible image fusion," *Information Fusion*, vol. 48, pp. 11–26, 08 2019.

[7] Moulay A. Akhloufi Axel-Christian Guei, "Deep generative adversarial networks for infrared image enhancement," 2018.

[8] Takeru Miyato, Toshiki Kataoka, Masanori Koyama, and Yuichi Yoshida, "Spectral normalization for generative adversarial networks," *arXiv preprint arXiv:1802.05957*, 2018.

[9] Alexia Jolicoeur-Martineau, "The relativistic discriminator: a key element missing from standard GAN," *CoRR*, vol. abs/1807.00734, 2018.

[10] Yungang Zhang and Yu Xiang, "Recent advances in deep learning for single image super-resolution," in *Advances in Brain Inspired Cognitive Systems*, Jinchang Ren, Amir Hussain, Jiangbin Zheng, Cheng-Lin Liu, Bin Luo, Huimin Zhao, and Xinbo Zhao, Eds., Cham, 2018, pp. 85–95, Springer International Publishing.

[11] Marc Bosch, Christopher M Gifford, and Pedro A Rodriguez, "Super-resolution for overhead imagery using densenets and adversarial learning," in *2018 IEEE Winter Conference on Applications of Computer Vision (WACV)*. IEEE, 2018, pp. 1414–1422.

[12] Wenzhe Shi, Jose Caballero, Ferenc Huszar, Johannes Totz, Andrew P. Aitken, Rob Bishop, Daniel Rueckert, and Zehan Wang, "Real-time single image and video super-resolution using an efficient sub-pixel convolutional neural network," in *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2016.

[13] Yuewen Sun, Litao Li, Peng Cong, Zhentao Wang, and Xiaojing Guo, "Enhancement of digital radiography image quality using a convolutional neural network," *Journal of X-ray Science and Technology*, vol. 25, no. 6, pp. 857–868, 2017.

[14] Chih-Yuan Yang, Chao Ma, and Ming-Hsuan Yang, "Single-Image Super-Resolution: A Benchmark," in *Computer Vision – ECCV 2014*, David Fleet, Tomas Pajdla, Bernt Schiele, and Tinne Tuytelaars, Eds., vol. 8692, pp. 372–386. Springer International Publishing, Cham, 2014.