Proceedings of 10th IOE Graduate Conference
Peer Reviewed
ISSN: 2350-8914 (Online), 2350-8906 (Print)
Year: 2021 Month: October Volume: 10

Perceptual Image Super Resolution Using Stacked Receptive field Blocks and U-Net

Surendra K C [a], Sharan Thapa [b]

[a, b] *Department of Electronics and Computer Engineering, Paschimanchal Campus, IOE, Tribhuvan University, Nepal*
Corresponding Email: [a] kcsurendra28@gmail.com, [b] sharant@ioepas.edu.np

Abstract
The low resolution images appear in many cases. Generative Adversarial Networks takes advantage of two independent neural networks to create realistic data. The estimation of a high resolution image from its counterpart low resolution image called super resolution which is used in this research using GANs. The input is a low resolution human face of size 64×64 is used which keeps certain information but not details. This network is capable of generation of images into 4× up scaling factors. The network is a min-max player game where the generator and the discriminator are trained simultaneously and competed against each other to reach the state where the discriminator is no more able to discriminate between the real and the fake image. This state is known as Nash Equilibrium. The main aim of this network model is to minimize the loss of the generator and maximize the loss of the discriminator so that the generator can generate more real looking images and the discriminator will be unable to differentiate between the image generated by the generator as the fake one. The use of Strong discriminator and the effective generator that is able to extract the coarse and excellent skin texture from Low Resolution input images with adversarial training containing VGG based perceptual loss can improve state-of-art of perceptual super resolution of Low Resolution images.

Keywords
Generative Adversarial Networks, Receptive Field Block, U-Net Discriminator, Super Resolutions

1. Introduction

The space of application that can be implemented with deep learning is nearly infinite. Super resolution (SR) is one of the applications of deep learning. It's the estimation of high resolution (HR) images from its equivalent low resolution (LR) images [1]. It is a really difficult task building neural networks that automatically reconstructs High Resolution images from its Low Resolution images. Surveillance, medical imaging, satellite image analysis, facial recognition, compressed picture improvement, and antique photo recovery are just a few of the applications of super resolution.

According to the number of input Low Resolution images, the Super Resolution can be divided into two groups, which are single image super resolution (SISR) and multi image super resolution (MISR) [2]. SISR is greatly admired due to its high effectiveness compared to MISR. SISR is an ill-posed problem since no one unique solution for a particular Low Resolution image exists. Example (learning)-based methods, interpolation-based methods, and reconstruction-based methods are the three primary types of SISR algorithms. Methods based on examples have seen tremendous growth in recent years due to their simple pipeline and amazing performance.

GANs (Generative Adversarial Networks) takes advantage of two independent neural networks to create realistic data, developed by Ian Goodfellow, then a PhD student at the University of Montreal in 2014 [3]. The generator and discriminator are the two networks that train each other through multiple cycles of generation and discrimination while also attempting to confuse one another. The generator is trained to create fake data, while the discriminator is trained to make a distinction between fake and actual data. The structural diagram of the GANs is shown in figure 1.

GANs offer a robust foundation for super resolution in accuracy and speed compared to Convolutions Neural Network (CNN) which were incapable of

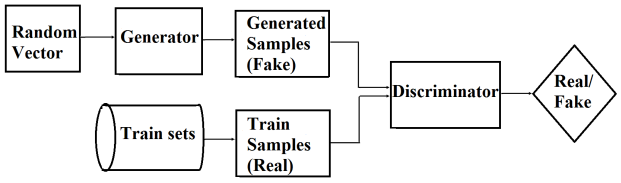

Figure 1: Generative Adversarial Networks (GANs)

recovering finer details and often generate blurry image in some case. GAN's optimization method is a min-max problem. The optimization process is completed when the generator and discriminator reach the Nash equilibrium. Any neural network may be used as a generator and discriminator, with the generator's function to generate realistic data and the discriminator as a classifier.

The need of super resolution and enhancement technique still cannot be ignored, while digital imaging devices with higher resolution have been rapidly developed in last decade.

High resolution images and videos are fundamental aspect for large displays like High Definition TV sets, large computer displays, and most recently the hand held smart phones. If the transmission network bandwidth is not sufficient for such devices, then visual pleasant from these devices cannot be achieved.

In this big data era, compression and Super Resolution may be the best approach to decrease network bandwidth use for services that provide streaming of high-quality multimedia data. The digital surveillance products sacrifice resolution to some degree for long term stable operation. For remote sensing, there is also a trade off between spatial, spectral, and temporal resolutions. Similar situation exist in medical imaging. Thus the need of super resolution has attracted attention for the computer vision research community for improving reconstruction details of the scenes and constituent objects in video surveillance, image/video streaming,remote sensing applications, and medical diagnosis.

The main goal of this paper is to develop generative adversarial network for perceptual super resolution of low resolution images, which can successfully retrieve texture and finer details from low resolution image and generate perceptual quality high resolution images.

2. Related Works

To tackle the SR problem in the realm of deep neural networks, Dong et al. [4] was pioneer and proposed SRCNN achieved a state-of-the-art result against the previous work. After then, the field has observer a number of network architecture. Still no method based on Deep Neural Network can achieve the best PSNR and the best quality at the same time [5]. More sophisticated techniques, which often rely on training data, seek to establish a complicated mapping between low and high resolution image information [1]. Many example-based approaches rely on low resolution training for which the matching high resolution is known Kim et al. [6] used deeply recursive convolution network (DRCN), which enables long-range pixel dependencies while keeping the model's parameter count modest. The use of LR image as a direct input led to significant reduction in computations, while the model's capacity and performance improvements are maintained.

Ledig et al. [1] introduce the SRResNet with skip connection, a deeper architecture made up of residual blocks for Low Resolution feature learning. This SRGAN model uses the adversarial with perceptual loss to prefer outputs based on a plethora of natural images. This model employs a weighted sum of content and adversarial loss to calculate perceptual loss. They conclude that the performance can be increase with the deeper network at the expense of longer training and testing periods. The result is stated on table 1.

Lim et al. [7] proposed EDSR model by eliminating unnecessary batch normalization layer in residual block as well as increasing the model's size, which achieve substantial progress. The batch normalization layer only introduces a shift to the feature and this shift may have an adverse effect on the ultimate performance. Removing Batch Normalization does not reduce performance but conserves memory and computational resources.

Wang et al. [8] improves the visual quality by improvising the SRGAN and proposed ESRGAN which achieves higher visual quality and more realistic results, that won PIRM2018 SR challenge [5]. In ESRGAN model batch normalization were removed from the generator network and the basic building block was Residual-in-Residual Dense Block (RRDB), which combines dense connections in a multi-level residual network. Beside the improved

generator the ESRGAN model uses the Relativistic GAN-based discriminator, which estimate the relative loss rather than absolute loss. The model tested on DIV8K data sets obtained PSNR 24.03dB and 0.54 SSIM.

Xining et al. [9] presented a GAN-based image super resolution with an unique quality loss based on the gradient magnitude similarity deviation IQA metric. This model is not perfect while dealing strong repetitive structures. And conclude that large quantity of high-resolution training pictures with a variety of textures can be benefits to generate visually appealing results. The result is stated on table 1.

OPPO-Research [10] proposed RFB-ESRGAN based on ESRGAN, which uses the multi-scale Receptive Fields Blocks (RFB) in the generator network to restore finer details and texture. RFB can extract the coarse and fine features from input LR images. RFB replace large kernels by several small kernels thus reduce the model as well as time complexity. The model tested on DIV8K data sets obtained PSNR 23.38dB and 0.5504 SSIM.

CLIPLAB [11] proposed to use GAN based on ESRGAN with learned perceptual similarity (LPIPS) loss instead of using VGG perceptual loss. The model uses U-Net structure discriminator to fully utilize the local and global content. They study the effect of LPIPS loss and conclude that better LPIPS does not always imply a higher visual quality and more performance gain can be expected by combining more effective generator architecture. The model tested on DIV8K data sets obtained PSNR 22.77dB and 0.5251 SSIM.

Edgar et al. [12] uses Generative Adversarial Networks with U-Net Discriminator, which allows providing feedback to the generator on a global and local scale. And this type discriminator gives the generator more detailed feedback. U-Nets have shown state-of-the-art performance in a variety of tasks. They found that U-Nets The discriminator can retrieve more information than the standard encoder architecture discriminator. The U-Net discriminator gives a more pronounced feedback to the generator than standard discriminator. Thus this strong discriminator makes the generator better.

3. Methodology

GAN is the main framework of this research, which includes generator and discriminator network. The methodology of the research is as in figure 2.

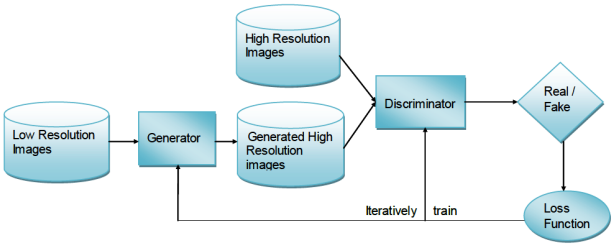

Figure 2: Work Flow of GAN Training.

3.1 The Generator

The generator network takes a Low Resolution image of a dimension of 64×64×3 and a series of convolution and up sampling generate a super resolution of a shape 256×256×3, while the discriminator examine High Resolution images and tries to figure out whether a given image is actual or not.

The objective of this research is to generate a super resolution image I^{SR} from a low resolution input image I^{LR}. The high resolution image I^{HR} is complement of the input low resolution image I^{LR} as

$$I^{LR} = d_\alpha I^{HR} \tag{1}$$

Where d_α is the degradation operation which when act on I^{HR} results in I^{LR} and α is the scaling factor and is less than 1. The task is to find an approximate inverse $f \approx d^{-1}$ to yield an High Resolution image estimate super resolution I^{SR} from I^{LR}. This problem is extremely difficult to solve as there exist a numbers of possible estimation of Super Resolution image I^{SR} for which the relation $I^{LR} = d_\alpha I^{HR}$ holds true.

The proposed generator network structure consists of convolution blocks, Receptive field block residual dense blocks (RFB-RDB), Receptive Field Blocks (RFB) and up-sampling blocks as in figure 3.

The first is convolution block of kernel size of 3×3 with filter size 64 followed by LeakyReLU as activation function with alpha 0.2. Then follows stacked of 16 RFB-RDB, each RFB-RDB contains five RFB and four LeakyReLU with alpha 0.2 as activation function in it as in figure 4.

Figure 5 depicts the composite structure of the Receptive Field Block. Receptive Field Block takes a

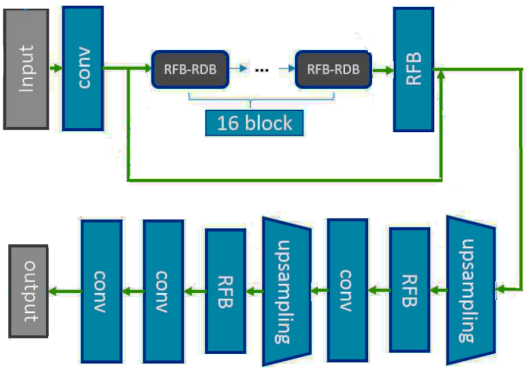

Figure 3: Generator Network.

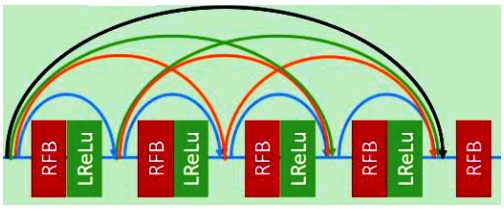

Figure 4: Receptive Field Blocks Residual Dense Block.

mixture of smaller kernels like (1×1, 1×3, 3×1), which has the ability to effectively decrease parameters and can extracts very details line features like hair, skin, edges texture etc which is needed in field of image reconstruction. The Receptive Field Block is composed of multi-branch convolution blocks with different kernels size and dilation convolution layers.

The first branch is 2D convolution layer with filter size of 64, kernel size 1. The second branch consists of two 2D convolution layers with filter size of 16, kernel size of 1 in first convolution and 3 in second convolution layers. The dilation rate of second convolution is 1. The third branch consists of three 2D convolutions with filter size of 16, kernel size 1 in first, (1, 3) in second and 3 in third convolution layers respectively. The dilation rate is 3 for the third convolution layers. The fourth branch consists of three 2D convolutions with filter size of 16, kernel size 1 in first convolution, (3, 1) in second convolution and 3 in third respectively. The dilation rate is 3 for the third convolution layers. The last branch consists of four 2D convolutions with filter size of 8, 12, 16 and 16 respectively, kernel size 1 in first, (1, 3) in second, (3, 1) in third and 3 in fourth convolution layers respectively. The dilation rate is 5 for the fourth convolution layers.

The second, third, fourth, and last branch are concatenate and scaled with 0.2 which is fed to

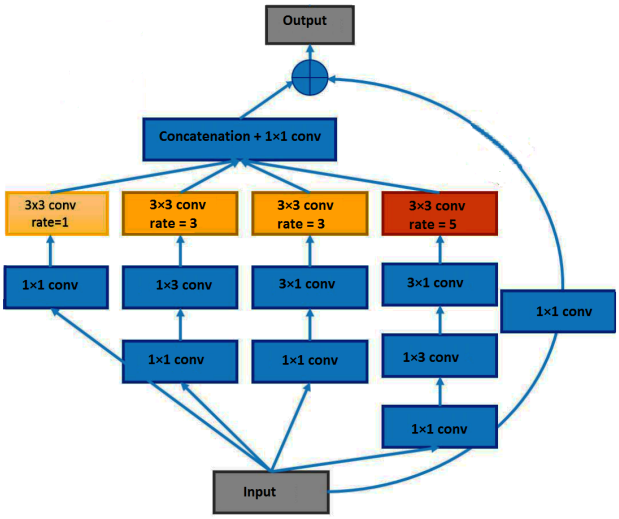

Figure 5: Receptive Field Block.

convolution block with 64 filters and kernel size of 1, which is added with first branch output. All convolution layers of each branches uses ReLU as activation function. The output of 16 stacked RFB-RDB is fed to a single Receptive Field Block block and two 2× up-sampling blocks. In up-sampling phase sub-pixel convolution is used, which makes space transformation to the RFB output. Each up-sampling block are followed by RFB and convolution block of kernel size 256 and filter size of 3. The computational complexity is also further decreases by reducing the number of parameters in sub-pixel convolution. Final a convolution block with filter size 3 and kernel size 3 with tanh as activation function is used.

3.2 The Discriminator

The encoder structured discriminator is used in the majority of GAN-based Super Resolution techniques. This type of discriminator focused on either global or local details. There has been study to improve the performance of images [12]. An encoder-decoder network popularly known as U-Net model is proposed for discriminator. The U-Net discriminator allows for both local and global data representation, giving the generator additional useful feedback. The proposed discriminator is as in figure 6.

The input of discriminator is the real/generated high resolution images of size 256×256×3. The U-Net is made up of down and up sampling networks linked by a bottleneck, as well as skip connections that replicate and concatenate the encoder's feature mappings with that of decoder. The first encoder consists of 2D

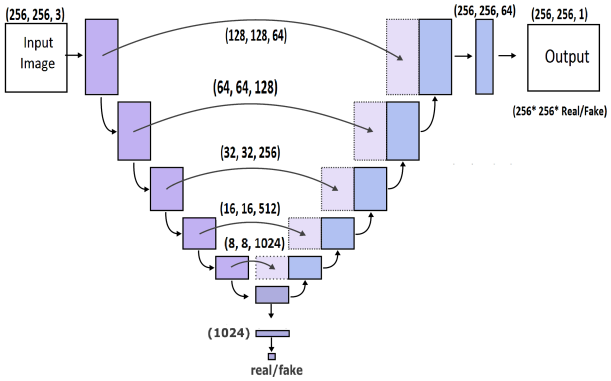

Figure 6: U-Net discriminator.

convolution block with filter size 64, kernel size of 3 and strides of 2. Similarly the second, third, fourth, fifth, and sixth encoder consists of 2D convolution with filter size of 128, 256, 512, 1024, and 1024 respectively with kernel size of 3 and strides of 2. After each convolution, there is a batch normalization layer with momentum 0.8 and an activation function called LeakyReLU with an alpha of 0.2. The output of sixth encoder is fed to dense layer of 1024, which is fed to dense units of 1 as encoder output for real and fake output using sigmoid activation function.

The decoder parts consist of serial 2× up-sampling blocks followed by 2D convolution, Batch Normalization, concatenation, and ReLU activation function. The first decoder consists of 2× up-sampling blocks with 2D convolution layer of 1024 filter size and kernel size of 3. The output is concatenated with fifth encoder output after batch normalization. After that, there's a ReLU activation function. Similarly the second decoder consists of 2× up-sampling blocks with 2D convolution layer of filter size 512 and kernel size of 3 followed by batch normalization. The output is concatenated with fourth encoder output which is followed with ReLU activation function. In the same manner third, fourth, and fifth decoder has 256, 128, and 64 filter 2D convolution with kernel size 3 after 2× up-sampling blocks and these outputs are concatenated with third, second, and first encoder after batch normalization respectively. All decoder have ReLU as activation function. The last 2× up-sampling blocks is followed by 2D convolution of filter size 1 and kernel size 3 with sigmoid as activation function. The decoder output is of size 256×256×1.

The U-Net discriminator D^U conducts per-pixel classification, segmenting the image into real and fake areas, as well as the encoder's original image classification. Thus the discriminator is enabled to learn local and global difference between fake and real images. The discriminator loss is computed by taking decisions form both encoder D^U_{enc} and decoder D^U_{dec} . The total loss is calculated as

$$L_{D^U} = L_{D^U_{enc}} + L_{D^U_{dec}} \tag{2}$$

The loss of encoder$L_{D^U_{enc}}$ is computed from the scalar output D^U_{enc} as

$$L_{D^U_{enc}} = -\mathbb{E}_x[\log D^U_{enc}(x)] - \mathbb{E}_z[\log[1 - D^U_{enc}(G(z))]] \tag{3}$$

The loss of decoder $L_{D^U_{dec}}$ is computed as mean decision over all pixels as

$$L_{D^U_{dec}} = -\mathbb{E}_x[\sum_{i,j} \log[D^U_{dec}(x)]_{i,j}] - \mathbb{E}_z[\sum_{i,j} \log[1 - D^U_{dec}(G(z))]_{i,j}] \tag{4}$$

$[D^U_{dec}(x)]_{i,j}$ and $[D^U_{dec}(G(z))]_{i,j}$ refer to discriminator output at pixel i,j.

3.3 Loss Function

To train the GAN network the objective function or loss function, which we need to minimize to train the model. The weighted sum of content and adversarial loss is the perceptual loss function, that is the objective function for this model.

Content loss is measuring the difference between generated image and the real image, which are of two types, VGG loss and Pixel-wise MSE loss. The most common way to calculate is the pixel-wise MSE loss. However, the MSE loss tends to generate overly smooth textures for output images which result in perceptually unsatisfying solution.

Thus pre trained VGG19 network functions is used as feature extractors, extracting features of generated images and real images. The perceptual VGG loss is defined as

$$L^{SR}_{VGG_{i,j}} = \frac{1}{W_{i,j}H_{i,j}} \sum_{x=1}^{W_{i,j}} \sum_{y=1}^{H_{i,j}} (\phi_{i,j}(I^{HR})_{x,y} - \phi_{i,j}(G_{\theta_G}(I^{LR})_{x,y})^2 \tag{5}$$

Where $W_{i,j}$ and $H_{i,j}$ are the dimensions of the feature maps within the VGG19 network, $\phi_{i,j}$ represents the feature map generated by the VGG19 network. It's the Euclidean distance between the produced image's feature maps and the real image's feature maps. And $G_{\theta_G}(I^{LR})$ represent the generated image.

The adversarial loss is calculated on the probability returned by the discriminator network. The discriminator is trained to distinguish the real image from generated image. Adversarial training is used to produce natural looking image. This adversarial loss is calculated as

$$L^{SR}_{Gen} = \mathbb{E}_Z[\log(D^U_{enc}(G(z))) + \sum_{i,j} \log[D^U_{dec}(G(z))]_{i,j}] \tag{6}$$

The perceptual loss utilized for this paper is a weighted average of VGG loss defined by equation 5 and adversarial loss defined by equation 6 as

$$L^{SR} = L^{SR}_{VGG} + 0.001 \times L^{SR}_{Gen} \quad (7)$$

4. Data Set

For dataset, CelebAMask-HQ [13] used, is a large face image data set containing 30,000 high-resolution face images selected from the CelebA data set according to CelebAHQ. The images of CelebAMask-HQ have the size of 512 x 512. The images for low resolution are chosen as 64×64 that keeps certain facial information. A random flip is done for data augmentation which create a mirror of original image. The images are converted to pixel values to the range between -1 to 1. The generator uses the tanh activation function that squashes the values to the same range.

5. Experimental Results and Discussion

5.1 Implementation details

The research is implemented in Google Colab, a product of Google research hosted on jupyter notebook. Keras is used as the high level python library that provides a convenient way to define and train deep learning architectures like Generative adversarial network through its functional API. Google's TensorFlow, is used as a backend library to perform low level array operation. The loss function stated in equation 7 & 2 was used to train the generator and discriminator respectively for 50K iterations with a mini-batch size of 1. The network was trained using the Adam optimizer with a learning rate of 0.0002. The generative network and discriminator network were alternatively updated.

5.2 Results and Discussion

The generator loss with image batch size of 1 for first 8,000 iterations is as in figure 7.

The loss of generator is high at beginning of the iteration and slowly lower with increasing iteration. This loss has variance.

The discriminator loss with image batch size of 1 for first 8,000 iterations is as in figure 8.

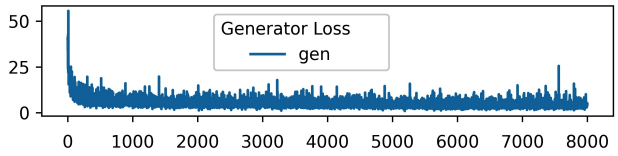

Figure 7: Generator Loss for Batch size 1.

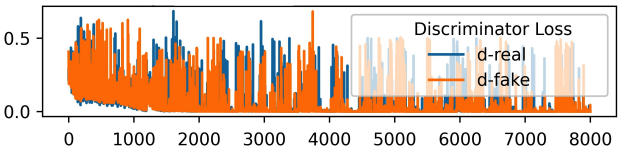

Figure 8: Discriminator Loss for Batch size 1.

The loss of discriminator for fake start with a high value and real with low. These losses increases slightly with increase in iteration and then decreases. These losses has variance. The loss of generator and discriminator have variance because they both are competing against each other, so if one gets better the other gets a larger loss.

Almost all deep learning model are compiled and trained to minimize loss. This is not true for GANs because smaller loss of generator doesn't means better quality of images. Because the generator loss is graded against the current discriminator which is constantly improving. The loss function evaluated at different points thus cannot be compared. Loss of generator may be high even image quality is improving.

The accuracy of discriminator to discriminate real and fake high resolution images with image batch size of 1 for first 8,000 iterations is as in figure 9.

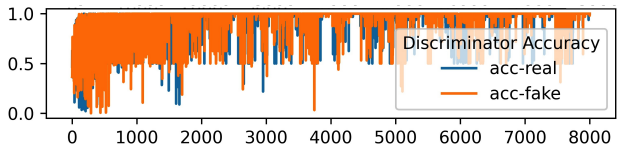

Figure 9: Discriminator Accuracy for Batch size 1.

The discriminator accuracy for classifying real and fake images should not remains high through out the run because it indicate that generator is poor at generating images in some way that it is easy for discriminator to identify fake images. The accuracy should hover around 70% to 80%.

The discriminator accuracy to detect real images starts at high while the accuracy to detect fake starts at low. The accuracy oscillate and become stable with variance at the end of iteration as in figure 9. When

the generator and discriminator are in a stable operation high quality images are generated and while during high variance loss lower quality images are generated.

The generator was trained with batch sizes of 4. The loss of generator with discriminator and discriminator accuracy curves for training with batch size of 4 are shown in figures 10 and 10.

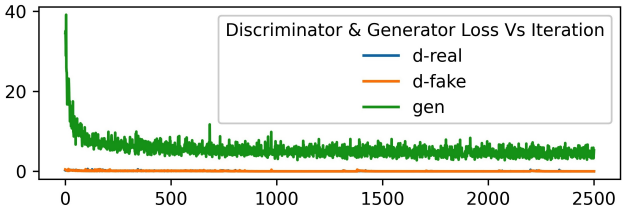

Figure 10: Generator and discriminator loss for Batch size 4

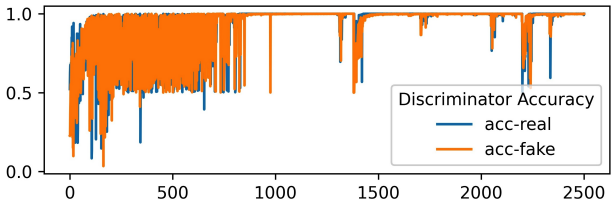

Figure 11: Discriminator accuracy for Batch size 4

With increasing batch size the the model is prone to convergence failure. In figure 10 the accuracy curves is almost at 100% most of the time after 900 iteration indicating that discriminator is perfect at identifying real and fake images.

Batch normalization, dropout, learning rate, activation layers, convolution filters, strides, batch size, and latent space are very sensitive parameters for GANs. Finding a set of these factors that works is frequently a subject of trial and error rather than following a set of established rules.

Training GANs is a very challenging and achieving a equilibrium is a trial and error method. Also perfect convergence of GANs is not defined. Both generator and discriminator learns from each other and constantly improving their capabilities. If one becomes more powerful at some time then the model becomes unstable. When discriminator is not powerful the randomly generated images may be passed as real which may be far from realness. Similarly when discriminator becomes more powerful no generated images will passed as real in spite the image very close to realness. For stable GANs the adversarial concept should be fulfilled.

5.3 Outputs

The trained model was tested on facial images from the Set5, Set14, and Nepali Portraits public benchmark data sets. The SSIM and PSNR is calcualted for each images. A higher scrore means a better result. The SSIM and PSNR for the test images are as in figure 12.

Figure 12: Input Low vs Output High resolution image.

Quantitative evaluation of proposed model with the SRGAN [1], GMGAN [9] on public benchmark datasets is as in table 1. A higher score means better quality. The average PSNR of this proposed model for the data sets stated in table 1 is 27.976dB and SSIM 0.830.

Although the human visual system are powerful at capturing and accurately assessing image quality than standard quantitative measures like SSIM and PSNR. The generated images looks sharper, real and perceptual in nature. From figure 12 it is clear that the generator have retrieve texture and finer details like lady's hat, the small flower branch in front of face, hairs and different facial parts present in low resolution images.

Table 1: Comparison with other models on Public benchmark test (PSNR[dB]/SSIM)

Model DATASET	SRGAN	GMGAN	Proposed Model
SET5	29.40/ 0.8472	30.02/ 0.8447	27.976/ 0.830
SET14	28.49/ 0.818	26.37/ 0.7055	
NEPALI PORTRAITS	-	-	

The images produced during the training of model and their corresponding feedback from U-net discriminator is as in figure 13.

The first column image of figure 13 is at 100 iterations and second column at 10,000 iterations and the last column is the original high resolution image and the bottom row is discriminator feedback on them.

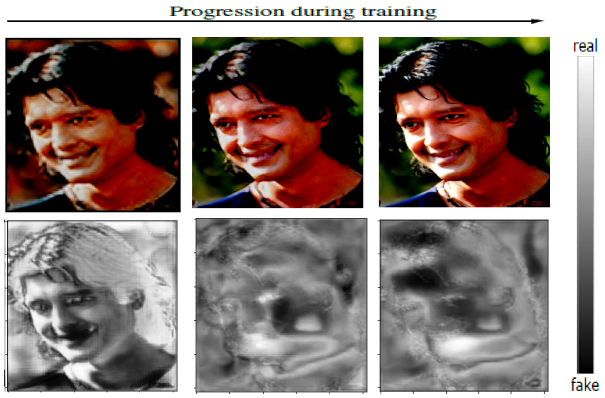

Figure 13: U-Net output during different training iteration.

Brighter pixel represents the assurance of the discriminator of that pixel as being real and darker as fake one.

The image is flip so that mirror image of original image is generated for data augmentation as shown in figure 14.

Figure 14: Image flip to create Mirror image.

6. Conclusion and Future Work

A generative adversarial network is used in this research is composed of stacked Receptive Field Blocks generator and U-Net discriminator was proposed. The receptive field blocks with small convolution kernel extract fine details of low resolution input image for reconstruction. Also U-Net based discriminator allows local and global feedback to generator, which is more informative.

Thus the use of Strong discriminator and the effective generator with adversarial training containing VGG based perceptual loss is able to extract the coarse and fine features from Low Resolution input images and generate perceptual super resolution images. The use of different loss function depends upon the application on which the model is used. The texture loss function is not considered in this research which might be helpful in reconstructing realistic texture to reduced visually outstanding artifacts, which is a part of future work.

References

[1] C. Ledig, L. Theis, F. Huszar, J. Caballero, A. Aitken, A. Tejani, J. Totz, Z. Wang, and W. Shi. Photo-realistic single image super-resolution using a generative adversarial network. In *Computer Vision and Pattern Recognition*, Jul 2017.

[2] Wenming Yang, Xuechen Zhang, Yapeng Tian, Wei Wang, Jing-Hao Xue, and Qingmin Liao. Deep learning for single image super-resolution: A brief review. *IEEE Transactions on Multimedia*, 21(12):3106–3121, 2019.

[3] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. *Advances in neural information processing systems*, 27, 2014.

[4] Chao Dong, Chen Change Loy, Kaiming He, and Xiaoou Tang. Image super-resolution using deep convolutional networks. *IEEE transactions on pattern analysis and machine intelligence*, 38(2):295–307, 2015.

[5] Yochai Blau, Roey Mechrez, Radu Timofte, Tomer Michaeli, and Lihi Zelnik-Manor. The 2018 pirm challenge on perceptual image super-resolution. In *Proceedings of the European Conference on Computer Vision (ECCV) Workshops*, pages 0–0, 2018.

[6] Jiwon Kim, Jung Kwon Lee, and Kyoung Mu Lee. Deeply-recursive convolutional network for image super-resolution. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1637–1645, 2016.

[7] B Lim, S Son, H Kim, S Nah, and KM Lee. Enhanced deep residual networks for single image super-resolution 2017 ieee conference on computer vision and pattern recognition workshops (cvprw), 2017.

[8] Xintao Wang, Ke Yu, Shixiang Wu, Jinjin Gu, Yihao Liu, Chao Dong, Chen Change Loy, Yu Qiao, and Xiaoou Tang. ESRGAN: enhanced super-resolution generative adversarial networks. *CoRR*, abs/1809.00219, 2018.

[9] Xining Zhu, Lin Zhang, Lijun Zhang, Xiao Liu, Ying Shen, and Shengjie Zhao. Gan-based image super-resolution with a novel quality loss. *Mathematical Problems in Engineering*, 2020, 2020.

[10] Taizhang Shang, Qiuju Dai, Shengchen Zhu, Tong Yang, and Yandong Guo. Perceptual extreme super-resolution network with receptive field block. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops*, pages 440–441, 2020.

[11] Younghyun Jo, Sejong Yang, and Seon Joo Kim. Investigating loss functions for extreme super-resolution. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops*, pages 424–425, 2020.

[12] Edgar Schonfeld, Bernt Schiele, and Anna Khoreva. A u-net based discriminator for generative adversarial networks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8207–8216, 2020.

[13] Shuo Yang, Ping Luo, Chen-Change Loy, and Xiaoou Tang. From facial parts responses to face detection: A deep learning approach. In *Proceedings of the IEEE international conference on computer vision*, pages 3676–3684, 2015.