Real-Time Single Image Super-Resolution using Dual Adversarial Learning

Supratik Banerjee B.Sc

A Dissertation

Presented to the University of Dublin, Trinity College

in partial fulfilment of the requirements for the degree of

Master of Science in Computer Science (Augmented and Virtual Reality)

Supervisor: Michael Manzke

August 2019

Declaration

I, the undersigned, declare that this work has not previously been submitted as an exercise for a degree at this, or any other University, and that unless otherwise stated, is my own work.

Supratik Banerjee

August 15, 2019

Permission to Lend and/or Copy

I, the undersigned, agree that Trinity College Library may lend or copy this thesis upon request.

Supratik Banerjee

August 15, 2019

Acknowledgments

I would like to thank my supervisor, Michael Manzke for allowing me the opportunity to research on my chosen topic, and for his valuable guidance and assistance whenever I needed it.

Secondly, I thank my family for their support, guidance and encouragement throughout the process.

Lastly, I would like to thank my friends for always being supportive and motivating me with co-operation and competition.

Supratik Banerjee

University of Dublin, Trinity College
August 2019

Real-Time Single Image Super-Resolution using Dual Adversarial Learning

Supratik Banerjee, Master of Science in Computer Science

University of Dublin, Trinity College, 2019

Supervisor: Michael Manzke

In this dissertation, real-time single image super-resolution using adversarial learning is investigated. With an extensive study of the literature, it is observed that not a lot of work has been put to explore the potential of real-time super-resolution using adversarial techniques. The primary objective of this research is to design a shallow network architecture that balances a good trade-off between reconstruction quality and inference time. To solve this objective, a network architecture is proposed using adversarial learning. This work has a two-fold contribution. Firstly, a novel generator architecture is proposed, which uses an iterative back-projection architecture, where the novel part is the use of a sub-pixel convolutional neural network instead of the standard deconvolution based network. Secondly, a novel dual adversarial network is proposed, which uses the above-mentioned generator network and two discriminators. The first discriminator network uses an image-based pixel-wise loss. The second discriminator uses a pixel-wise loss on its feature maps. The novel part about the network is the use of a Relativistic GAN for both, instead of a Standard GAN. Overall the presented model demonstrates very promising results, but still suffers due to lack of hyper-parameter tuning. The sour code of this dissertation is available online: https://github.com/supratikbanerjee/SuperResolution

Contents

List of Tables

List of Figures

Chapter 1

Introduction

1.1 Overview

The gaming industry today has a massive demand for physically accurate rendered content. This can be observed by the race between Nvidia and AMD to push the boundaries of real-time rendering with their upcoming line of GPUs [7]. With the launch of the Nvidia RTX line of GPUs, they have taken the lead in the market by enabling real-time ray-tracing [8]. This has lead to pushing the limits of computational power for real-time rendering. Enabling ray-tracing significantly affects the rendering capability of the GPU, leading to drops in frame rate. As a solution, Nvidia proposed Deep Learning Super-Sampling (DLSS), which renders the frames at a lower resolution and then upscales it using Deep Learning techniques. These Deep Learning models are trained on Nvidia's Saturn V super-computing cluster [9]. A significant component of such a technique is called super-resolution. Since this technology offered by Nvidia is proprietary, there isn't much available information. Reaching out to the researchers at Nvidia didn't provide much information but helped in getting a direction to this research with their insight.

Single image super-resolution (SISR) has piqued interest in the computer vision research community, particularly since the pioneering work of Dong et al. (SRCNN 2014) [10]. This work has been followed by numerous propositions of new architectures and training strategies using Deep Convolution Neural Network approaches, demonstrating promising results. Given the advances in such accuracy of single image super-resolution, it remains a challenge to deploy these models in real-world applications. This is mainly due to the size of the models resulting from very deep networks constituting to high parameter count, thus increasing computational cost [2]. A possible solution to this problem would be continuing research towards designing lightweight deep networks. This suggestion by Yang et al. [2] was in line with the insight gained from Nvidia researchers, where they spoke about using a "tiny mode" for image super-resolution in DLSS.

In this research, the focus is around designing a lightweight network architecture factoring minimal reduction in quality. With this in mind, several network architectures have been analyzed to understand their fundamental principals of network design. In this work, a novel super-resolution network architecture is proposed, which is designed in two-fold. Firstly, the generator network is designed by comparing the deconvolution layer, and the sub-pixel convolution layer. Following a theoretical analysis presented by Shi et al. [11]. It is hypothesized that the sub-pixel convolutional layer with higher parameters is superior to the deconvolution layer, as it performs at the same speed. Thus sub-pixel convolutional networks have greater representation power. This hypothesis is experimentally proven to conclude the appropriate design choice. Following this, the generator network incorporates a novel back-projection [1] block basing on the first hypothesis. Secondly, a novel adversarial training process is introduced. Along with an image-based perceptual loss, a structural feature-based perceptual loss is also used. The proposed adversarial learning uses a Relativistic average Discriminator [12] to minimize the image-based and feature-based loss. The intuition behind using the feature-based perceptual is to use better image structural representation, as feature maps represent an image's structural characteristics. This, coupled with a Relativistic GAN [12] which uses a non-saturating loss, is easier to train.

The results of this research demonstrate superior image reconstruction in comparison with other super-resolution techniques with very few parameters in real-time. Additionally, by increasing the parameters of the network, the proposed generator achieved state-of-art-results. Furthermore, the proposed adversarial learning method demonstrates significant improvements in comparison with other adversarial techniques. The results evaluated in this work are only on 2x upscale, since for gaming the required quality of final image is desired to be really high. Thus any scaling factor above 2x would not be appropriate.

1.2 Motivation

The motivation for this research bases on the increasing popularity of using Deep Learning techniques for real-time rendering. In particular, with the emergence of DLSS, which uses image super-resolution to advance real-time rendering in modern games. This is coupled with the extensive literature available on single image super-resolution. With a thorough literature review, it was observed that very few are targeted towards achieving real-time performance. Some of the most inspiring work in literature that led to the contributions in this research include the work by Haris et al. [1] on the Back-projection networks. They propose allowing the model to learn the degradation process along with the upsampling. Li et al. [5] discuss a very crucial point of making use of fewer parameters to avoid overfitting. Their work achieves this by using a Back-projection block with an RNN like structure. Furthermore, Ledig et al. [13] firstly proposed the use of a Generative Adversarial Network (GAN). Their work demonstrated the limitations of PSNR and SSIM and proposed a solution by using perceptually based loss. Based on the concept of Relativistic GAN proposed by Jolicoeur-Martineau [12], Wang et al. [6] demonstrate the superiority of the idea over Standard GANs [14] in image super-resolution.

Chapter 2

Background and Related Work

2.1 Super-Resolution

Image super-resolution (SR) is a classical problem in computer vision [15]. It is the process of recovering one or more high-resolution (HR) images, given one or more low-resolution (LR) images [16]. It is defined as an inverse problem that combines de-noising, de-blurring, and scaling-up tasks to obtain high-quality signals from degraded ones [17]. Generally, the LR input I_{LR} is modeled as the output of some degradation process $\mathcal{D}$:

$$I_{LR} = \mathcal{D}\left(I_{HR}; \psi\right), \tag{2.1}$$

where I_{HR} is the Ground Truth image and ψ is the parameter for the degradation process. Usually, the degradation process $\mathcal{D}$ and parameter ψ is unknown and can be affected by several factors (defocusing, compression artifacts, sensor noise, etc.). The degradation mapping in most work is directly modeled as:

$$\mathcal{D}\left(I_{HR}; \psi\right) = \left(I_{HR}\right) \downarrow_s, \{s\} \subset \psi \tag{2.2}$$

where $\downarrow_s$ is the downsampling operation with scaling factor s. Other works [18], also model the degradation as a combination of multiple operations:

$$\mathcal{D}\left(I_{HR}; \psi\right) = \left(I_{HR} * \kappa\right) \downarrow_s + n_\sigma, \{\kappa, s, \sigma\} \subset \psi \tag{2.3}$$

where $I_{HR} * \kappa$ represents the convolution between the blur kernel κ and the HR image I_{HR}, and n_σ is some additive Gaussian noise with standard deviation σ. Comparing the definition of degradation in 2.2, the definition of degradation in 2.3 is closer to real-world use cases [18].

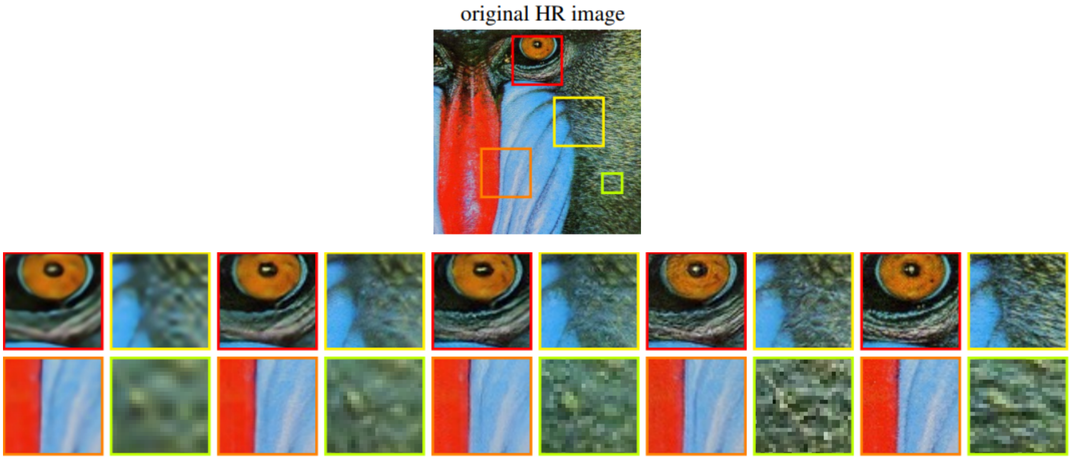

Figure 2.1: Super-Resolution as an ill-posed problem: Multiple solutions to HR image.

Solving (2.1) is an extremely ill-posed problem since there can be multiple HR solutions corresponding to given LR image [Figure 2.1]. Given this ill-posed nature of super-resolution, regularization is needed to constrain the solution. The primary objective of SR is:

$$\hat{\phi} = \underset{\phi}{arg\,min} \mathcal{L}\left(I_{SR}, I_{HR}\right) + \lambda\tau(\phi) \tag{2.4}$$

where $\mathcal{L}\left(I_{SR}, I_{HR}\right)$ is the loss function $\tau(\phi)$ is the regularization term (prior) and λ is the trade-off parameter. Super-Resolution provides details finer than the sampling grid by increasing the pixel count per unit area [17]. This involves recovering high-resolution image I_{SR} from LR input I_{LR} by reversing the above degradation process, such that, the recovered high-resolution image I_{SR} is identical to the ground truth I_{HR}. Therefore the inverse process of degradation can be formulated as:

$$I_{SR} = \mathcal{G}\left(I_{LR}; \phi\right), \tag{2.5}$$

where $\mathcal{G}$ is the super-resolution function, and ϕ is the parameter for the process.

Based on the input count of LR images, super-resolution can be classified into single image super-resolution (SISR) and multi-image super-resolution (MISR). SISR is much more popular in comparison with MIMR because of its high efficiency [2].

2.2 Deep Learning in Super-Resolution

Deep Learning (DL) is a sub-field of Artificial Neural Networks (ANNs) [19], which learns a diverse representation of data, and its performance is dependant on the choice of data representation [20]. In contrary to traditional task-specific learning algorithms, which take advantage of feature engineering with expert knowledge, deep learning aims to make learning algorithms less dependant on handcrafted features [20] and learn hierarchies of abstract data representation [19].

In literature, several single image super-resolution models with Deep Learning have been proposed [2]. Most of the work focuses on supervised super-resolution. In supervised SR, the models are trained with both the LR input and ground truth HR image. The various models proposed vary a lot in their intuition and implementation. Even then, its mostly combinations of various components, such as upsampling techniques, network design choice, and learning strategies. This section focuses on analyzing the fundamental modular components of various SISR models and summarizing their pros and cons.

2.2.1 Types of Super-Resolution using Deep Networks

The ill-posed nature of image super-resolution gives rise to the key question, how to perform upsampling? Although the architectures of different SR models vary a lot, they can be divided into four types [1].

2.2.1.1 Pre-Upsampling

Learning the mapping from low-dimensional space to high-dimensional space is a difficult problem. An naive solution to this is to use a traditional upsampling algorithm such as interpolation and obtain a higher resolution image, also referred to as middle resolution (MR) [1]. This MR image can then be used to learn MR-to-HR mapping

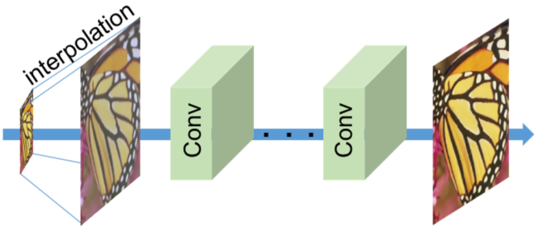

Figure 2.2: Pre-Upsampling. [1]

with Deep Neural networks. This was first introduced by Dong et al. in SRCNN [10].

The advantage of using pre-sampling is that the traditional algorithms handle the difficult part of upsampling the image and DNNs only need to refine the information in the MR image, which reduces the learning difficulty. Further, improved models [21, 22, 23] exploited different posterior designs and learning strategies.

Even though predefined upsampling reduces the learning difficulty, it might produce new noise or blurring from the MR image. Also, since most of the operations are executed in a high-dimensional space, the cost of time and space is higher than other methods.

2.2.1.2 Post-Upsampling

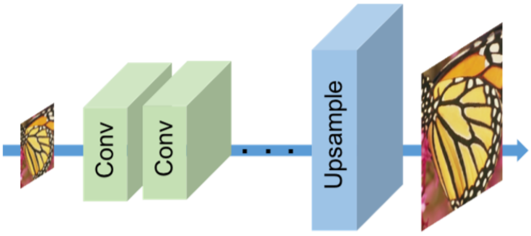

Figure 2.3: Post-Upsampling. [1]

To resolve computational inefficiency and use the full potential of representational learning for super-resolution, Dong et al. [24] and Shi et al. [25] proposed performing most of the mappings in the low-dimensional space. This method replaces the predefined upsampling operators at the beginning with end-to-end learnable upsampling layers at the end of the model. Since the expensive feature extraction via nonlinear convolutions occurs in the low-dimensional space and the resolution is increased at the end of the network, the cost of time and space is significantly reduced making training and inference considerably faster.

These advances make this method one of the most mainstream technique in the field of image super-resolution [13, 26, 27, 5, 4, 28, 29]. However, FSRCNN [24] and ESPCN [25] fail to learn complicated mapping due to limit capacity of the network. EDSR [30], belongs to this type, but it requires a high number of filters per layer. These problems opportunities to propose shallower networks that can preserve the high frequency details better [1].

2.2.1.3 Progressive Upsampling

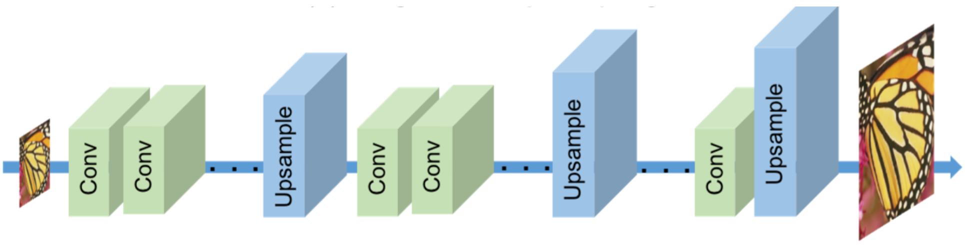

Figure 2.4: Progressive Upsampling. [1]

Even though models using pos-upsampling method have signifacntly reduced the large cost in terms of time and space, it still suffers from some shortcomings. In post-upsampling, since the upsampling operation is performed at the end, it significantly increases the learning difficulty for large scaling factors (e.g., 4, 8). At the same time, each scaling factor requires a separate super-resolution model. To address these drawbacks and support multi-scale SR a progressive upsampling SR method is proposed by Lai et al. [31]. In this method, the model is constructed in blocks, where in each block the HR image is progressively reconstructed to a higher scaling factor than the previous layer. In [32], Lai et al. improved on their method by using deep and wider recursive architecture. Progressive SR (ProSR) [33] proposed by Wang et al. also uses this method and achieve relatively better results.

By subdividing a complex task into small simpler tasks, this method significantly reduces the models learning difficulty. It not only obtains better performance with large scaling factors, but also provides an elegant solution to multi-scale SR problem without adding extensive cost in terms of time and space. Due to its multi-stage design, this method can further reduce learning difficulty and render better results by incorporating learning strategies like curriculum learning [34].

2.2.1.4 Iterative Up and Down Sampling

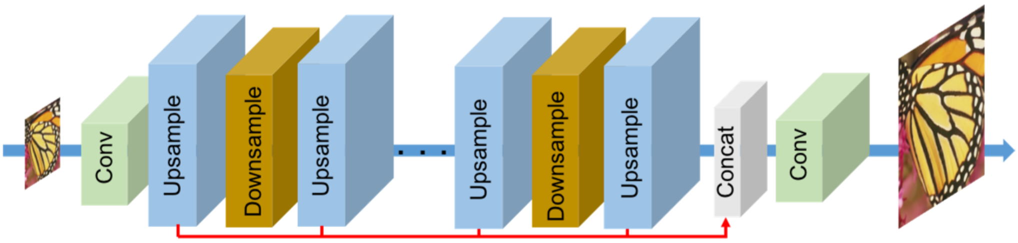

Figure 2.5: Iterative Up and Down Sampling. [1]

To better capture the mutual relation between LR and HR image pairs an iterative back-projection [15] is used. In this method, the reconstruction error is iteratively computed and fused back to tune the HR image intensity. Previous work on back-projection did not employ deep learning. Haris et al. introduced Deep Back-Projection network (DBPN) [1] which makes use of densely connected [35] up-sampling and down-sampling layers, refered to as up-projection and down-projection units alternatingly and reconstructs the HR image by concatenating the HR feature maps from all up-projection blocks. Likewise, Li et al. proposed Feedback Network [36] based super-resolution (SRFBN) [5], which uses a simplified version of densely connected up-sample and down-sample layers, which demonstrate similar performance to that of DBPN. The design principal of back-projection is still very new in lights of deep learning, and needs further exploration

2.2.2 Upsampling Methdods

Besides where to apply the upsampling operators in a model, it is also significantly important, how to implement them. Recently it has become a trend [10, 21, 24, 25, 30, 13, 33, 4, 28, 1, 22, 23, 26, 5, 29, 31, 32, 6] to make use of neural networks to learn upsampling process in an end-to-end fashion, despite having several traditional upsampling algorithm [37, 38, 39, 40]. This section will explore some of the commonly used deep-learning based upsampling methods.

Deep Learning-based Upsampling

Interpolation based upsampling methods increase the image resolution only based on the self-contained information. In the process, they might add noise or blur to the super-resolved image. To overcome these limitations of interpolation-based methods in the field of super-resolution and to learn the upsampling process in an end-to-end fashion, the deconvolution layer [24] and sub-pixel layer [25] are introduced.

Deconvolution Layer:

The Transposed Convolution layer was proposed by Zeiler et al. [41]. It performs an inverse transform of the normal convolution layer. For convolution, if the filter is convolved with an image with stride s then the approximate output is $1/s$ times the input size. In an inverse manner, if we swap the input and output, the output of deconvolution will be s times the input. This way, it aids super-resolution by expanding the image by setting the scaling factor equal to the stride of the deconvolution. Upon expanding the image, it fills the missing information with zeros. This expanded zero-filled image is then convolved with a $k \times k$ kernel. This way the input feature map is upsampled by a factor of s [Figure 2.6]. Since the deconvolution layer can upscale an image in an end-to-end fashion, it is used as the upsampling layer in many SR models [5, 26, 27, 42]. Despite its advantages, deconvolution can cause "uneven overlapping" on each axis [43] and hurt the SR results.

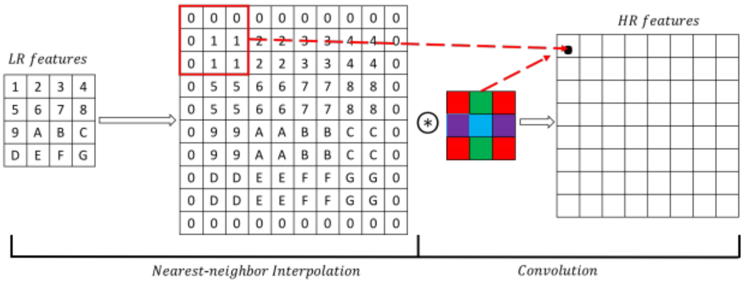

Figure 2.6: Deconvolution Layer [2]

Sub-Pixel Layer:

The sub-pixel layer introduced by Shi et al. [25]. It performs upsampling by generating several channels by convolution and then reshaping them using its ”shuffle” operation. In this layer, first, convolution is performed to generate s^2 times channels as the output size, where s is the scaling factor. Given the input size as $h \times w \times c$, the output size will be $h \times w \times s^2 c$. Performing the shuffle operation produces an output of size $sh \times sw \times c$ [Figure 2.7].

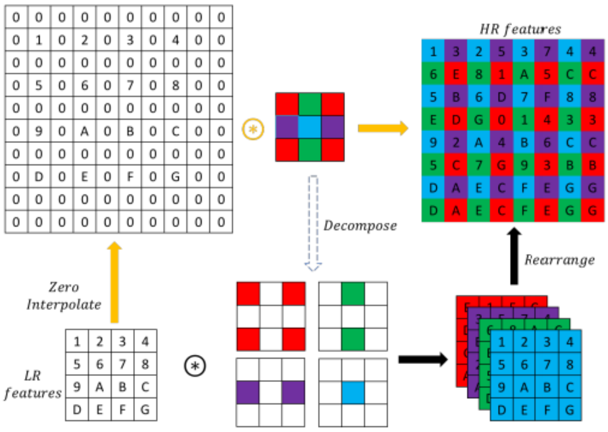

Figure 2.7: Sub-Pixel Layer [2]

A sub-pixel layer is also an end-to-end learnable upsampling layer, which has made this as well a good choice as an upsampling layer in several SR models [13, 44, 28, 4, 29].

These learning-based layers are currently the most popular choice for any network architecture using the post-upsampling method. This is primarily due to the fact that these layers avoid the use of a high-dimensional space for high-level feature extraction, thus remaining relatively computationally inexpensive.

2.2.3 Network Design

In deep learning, the choice of network architecture plays a very crucial role. Various network design strategies are combined with the four types of upsampling methods discussed above to formulate the final network architecture. This section discusses the granular principals of network design.

2.2.3.1 Residual Learning

Residual learning is a very popular choice of learning strategy adopted by various super-resolution models [45, 38, 46]. The primary objective to use this strategy is to alleviate the problem of degradation [Figure 2.8], where with increase in depth of the network, which is equivalent to adding more layers in a model, demonstrates higher training errors [47].

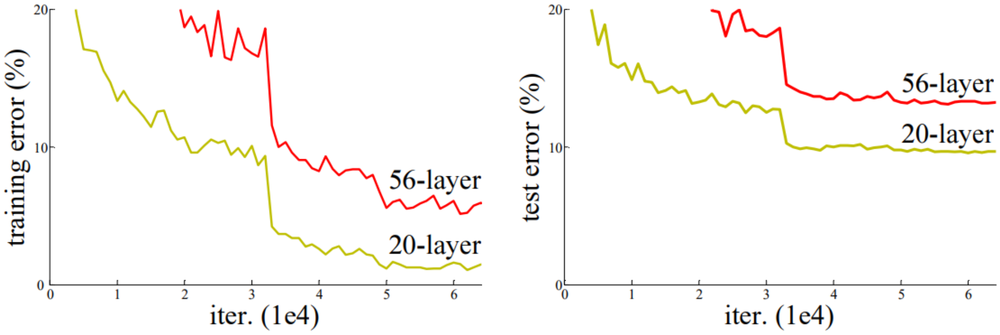

Figure 2.8: Degradation: Deeper Network shows higher train error, thus test error. [3]

Let $\mathcal{H}(\mathbf{x})$ be an underlying mapping "between LR and HR images" to be fit by few stacked layers, with $\mathbf{x}$ representing the input to the first layer. If we assume that these stacked layers can approximate the complicated mapping $\mathcal{H}(\mathbf{x})$, likewise we can state that the stacked layers can approximate the residual function $\mathcal{H}(\mathbf{x}) - \mathbf{x}$. Instead of approximating $\mathcal{H}(\mathbf{x})$, these layers are made to explicitly approximate a residual function $\mathcal{F}(\mathbf{x}) := \mathcal{H}(\mathbf{x}) - \mathbf{x}$. Thus the function can be reformulated as $\mathcal{F}(\mathbf{x}) + \mathbf{x}$ [Figure 2.9] to address the degradation problem.

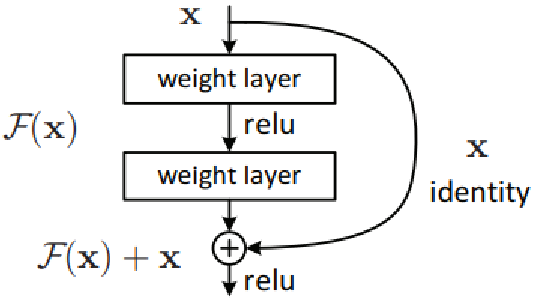
x
weight layer
relu
weight layer
$\mathcal{F}(\mathbf{x})$
x
identity
$\mathcal{F}(\mathbf{x}) + \mathbf{x}$
relu

Figure 2.9: Residual Learning Block [3]

Furthermore, Residual Learning can be classified into two categories, namely Global Residual Learning and Local Residual Learning.

Global Residual Learning:

Super-Resolution is a problem, where the input LR image is highly correlated with the target high-resolution image as the low-resolution image comprises abundant low-frequency information [4]. To avoid learning complications, the global residual learning is employed where the network only learns the residual between two images to restore the high-frequency information. It is adopted in several network architectures [21, 23, 4, 48].

Local Residual Learning:

As we construct deeper networks, we land in the problem of vanishing gradients [49] increasing the difficulty to train, leading to degradation [50]. To address this issue, He et al. [3] proposed the residual skip connection. Formally it can be expressed as:

$$I_l = \mathcal{H}_l\left(I_{l-1}\right) + I_{l-1}, \tag{2.6}$$

In Local Residual Learning, there are multiple shortcut connections between layers to alleviate the vanishing/exploding gradient problem. It is widely adopted by SR models [4, 48, 51, 23].

2.2.3.2 Dense Connections

The DenseNet proposed by Huang et al. [35] has demonstrated very promising results in various vision based tasks [52, 53, 54, 55, 56]. In a dense connection, every layer uses the feature maps from all previous layers [Figure 2.10]. The ℓ^{th} layer receives feature-maps of all preceding layers, $\mathbf{x}_0, ..., \mathbf{x}_{\ell-1}$ as inputs:

$$\mathbf{x}_\ell = \mathcal{H}_\ell\left(\left[\mathbf{x}_0, ..., \mathbf{x}_{\ell-1}\right]\right), \tag{2.7}$$

where $[\mathbf{x}_0, ..., \mathbf{x}_{\ell-1}]$ represents concatenation of feature-maps from layers $0, ..., \ell - 1$.

This structure not only helps to resolve the vanishing/exploding gradient problem but also significantly reduce parameters by using small growth rate [35]. Since

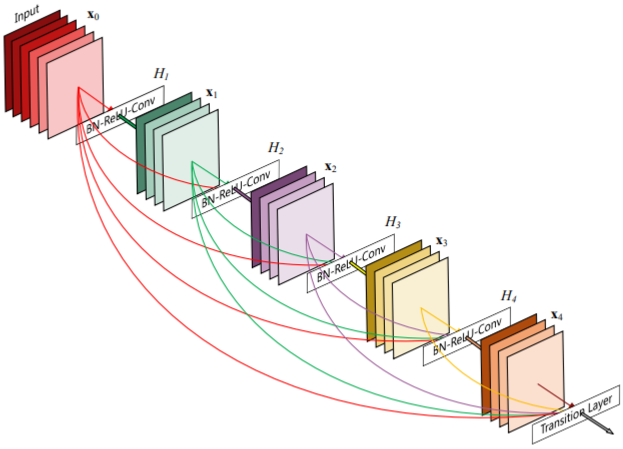

Figure 2.10: Densely Connected Convolutional Network

DenseNet allows reuse of features by introducing direct connections from any layer to all subsequent layers along with its other advantages, it has become widely popular in several super-resolution models [44, 1, 51, 28, 6, 33, 26, 57].

2.2.3.3 Channel Attention

Attention in context to deep learning can be viewed as allocating higher computational resources towards most informative components [58, 59]. Attention mechanisms have demonstrated promising results in tasks of understanding in images [60, 61], image captioning [62] etc. Hu et al. proposed [63] ”Squeeze-and-Excitation” block, where the focus is on the interdependence of feature representations between different channels. It demonstrates that the squeeze-and-excitation block significantly improves performance in state-of-the-art CNNs at a minor additional computational cost.

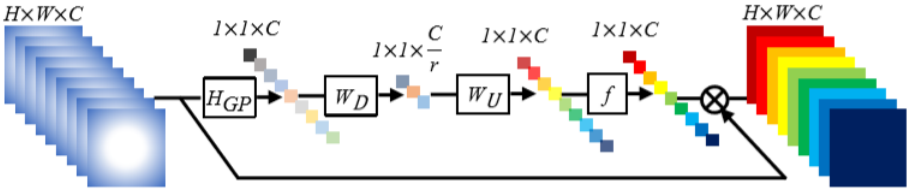

Figure 2.11: Channel Attention [4]

In this block, channel-wise global spatial information is put into a channel descriptor, usually by using a global average pooling. This phase is squeezing, where an input $\mathbf{X} = [\mathbf{x}_0, ..., \mathbf{x}_C]$ with C feature maps of size $H \times W$ is shrunk through spatial dimension $H \times W$ to derive the channel-wise statistic $\mathbf{z} \in \mathbb{R}^C$. The c^{th} element of $\mathbf{z}$ is defined as:

$$z_c = \mathcal{F}\left(\mathbf{x}_c\right) = \frac{1}{H \times W} \sum_{i=1}^{H} \sum_{j=1}^{W} \mathbf{x}_c(i, j), \tag{2.8}$$

where $\mathcal{F}(.)$ denotes the global average pooling function and $\mathbf{x}_c(i, j)$ denotes position of the c^{th} feature. Contribution from such descriptors expresses the whole image [63]. Further, the aggregated information in the squeeze operation is used to capture channel-wise dependencies. For this purpose, a gating mechanism is introduced with sigmoid activation.

$$s = \sigma\left(W_U \delta\left(W_D z\right)\right), \tag{2.9}$$

where $\sigma(.)$ denotes sigmoid and $\delta(.)$ denotes ReLU [64] activation. $W_D \in \mathbb{R}^{\frac{C}{r} \times C}$ and $W_U \in \mathbb{R}^{C \times \frac{C}{r}}$ are the weight set of the downsampling and upsampling convolutional layers, which are part of the bottleneck layer, i.e for dimensionality-reduction layer, followed by ReLU and a dimensionality-increasing layer with ratio r. s is the channel statistic obtained, which is used to rescale the inputs $\mathbf{X} = [\mathbf{x}_0, ..., \mathbf{x}_C]$:

$$\hat{\mathbf{x}_c} = s_c . \mathbf{x}_c, \tag{2.10}$$

where s_c is the channel statistic scaling factor and $\mathbf{x}_c$ is the c^{th} channel feature map.

Recently, Zhang et al. proposed RCAN [4] demonstrating the use of channel attention in super-resolution, which renders state-of-the-art results. As argued in [63] and [4], there can be better aggregation technique, Dai et al. demonstrated a Second-order Attention Network [29] using co-variance normalization and global co-variance pooling instead of global average pooling.

2.2.4 Loss Functions

Loss functions play a critical role at training deep learning models, as they act as an end state feedback to the network by providing a quantitative measure of the difference between the current learning state from the required learning state by assessing against the target variables. In the same light, loss functions are used to measure the difference between the super-resolved image and the ground truth image, guiding the model optimization.

Pixel Loss:

Pixel-wise loss employs measuring the difference between the HR image and SR image in a pixel by pixel order. At the pioneering stage of this research field with deep learning [10, 21, 25, 24], using pixel-wise L2 loss (mean squared error) was a common practice.

$$\mathcal{L}_{pixel-\ell2}(I_{SR}, I_{HR}) = \frac{1}{H \times W \times C} \sum_{x}^{W} \sum_{y}^{H} \sum_{i}^{C} \left\| \mathcal{G}\left(I_{LR}; \phi\right)^{x,y,i} - I_{HR}^{x,y,i} \right\|^{2}, \qquad (2.11)$$

This resulted in typically overly smooth textures rendered poor perceptual quality [65, 66]. Investigating the difference between mean squared error (MSE) and mean absolute error (MAS) to optimize Neural Networks for Image Restoration, Zhao et al. [67] demonstrated faster convergence and better results with mean absolute error. A variant of L1 loss used in [31, 68] is the Charbonnier loss [69].

$$\mathcal{L}_{pixel-\ell1}(I_{SR}, I_{HR}) = \frac{1}{H \times W \times C} \sum_{x}^{W} \sum_{y}^{H} \sum_{i}^{C} \left\| \mathcal{G}\left(I_{LR}; \phi\right)^{x,y,i} - I_{HR}^{x,y,i} \right\|, \qquad (2.12)$$

$$\mathcal{L}_{pixel-char}(I_{SR}, I_{HR}) = \frac{1}{H \times W \times C} \sum_{x}^{W} \sum_{y}^{H} \sum_{i}^{C} \sqrt{\left\| \mathcal{G}\left(I_{LR}; \phi\right)^{x,y,i} - I_{HR}^{x,y,i} \right\| + \epsilon^{2}}, \qquad (2.13)$$

where ϵ is a constant used for numerical stability.

The definition of PSNR is highly correlated with pixel-wise difference, hence minimizing pixel loss maximizes PSNR. Since pixel loss doesn't take perceptual [66] or texture [70] quality into account; thus, it lacks high-frequency details producing perceptually unsatisfactory results.

Content Loss:

Given the limitations seen with PSNR, optimization based on perceptual quality loss have been investigated [66, 65]. The intuition behind the perceptual loss is to use a pre-trained network to transfer the knowledge of semantic difference from the network loss to the model. The content loss is defined as the euclidean distance between feature representation of super-resolved image $\mathcal{G}\left(I_{LR};\phi\right)$ and the ground truth I_{HR} [13]:

$$\mathcal{L}_{\text{ content}}\left(I_{SR}, I_{HR};\Psi,\ell\right)=\frac{1}{W_{\ell}H_{\ell}C_{\ell}}\sum_{x}^{W}\sum_{y}^{H}\sum_{i}^{C}\left(\Psi_{x,y,i}^{(\ell)}(\mathcal{G}\left(I_{LR};\phi\right))-\Psi_{x,y,i}^{(\ell)}(I_{HR})\right)^{2}, \tag{2.14}$$

where Ψ is the pre-trained network and its feature map obtained at ℓ^{th} layer is defined as $\Psi^{(\ell)}$. As opposed to the pixel loss, the content loss doesn't force the super-resolved image I_{SR} to match exactly by pixels, but produce images perceptually similar to the target I_{HR}. This produces visually pleasing results [66, 13, 6, 71]. The most commonly used such pre-trained network is the VGG net [72].

Adversarial Loss:

Recently GANs [14] have picked up good traction in various vision based tasks. GANs employ a game theory based approach, where two components of the model, the generator and the discriminator compete against each other.The generator generates the content, in the case super-resolved images and the discriminator which takes the generated output and the target images as input and discriminates whether a given input is from the target distribution. As we can observe from the equation [14]:

$$\min_{G}\max_{D}V(D,G)=\mathbb{E}_{\boldsymbol{x}\sim p_{\text{data}}(\boldsymbol{x})}[\log D(\boldsymbol{x})]+\mathbb{E}_{\boldsymbol{z}\sim p_{\boldsymbol{z}}(\boldsymbol{z})}[\log(1-D(G(\boldsymbol{z})))] \tag{2.15}$$

where G and D are the Generator and Discriminator networks. Here the objective of the Generator network $G(\boldsymbol{z})$ is to generate a tensor that has the shape of the target data and follows its distribution. The function of discriminator network $D(\boldsymbol{x})$ is to output a scalar value between 0 and 1, which is the probability of whether a given input is from the target dataset or not. Given this context, it can be observed that the training process takes place in two steps. The first part keeps G constant and maximizes (2.16). As we can see in [Figure 2.12] as $\lim_{x\to 1}f(x)=0$ and $\lim_{x\to 0}f(x)=-\infty$, likewise with

$D(\boldsymbol{x}) \to 1$ maximizes the first term of the equation.

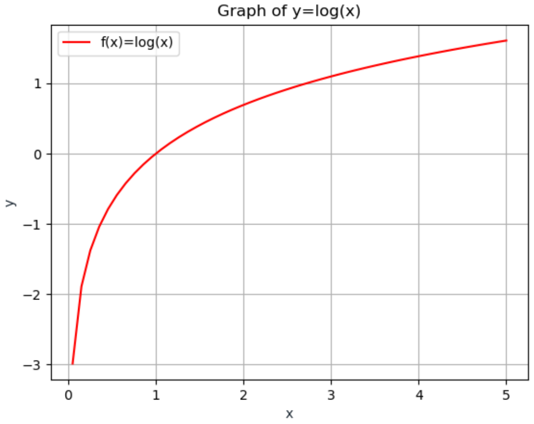

Figure 2.12: log(x) plot

The second term of the equation is maximized as $D(G(\boldsymbol{z})) \to 0$. This represents that to maximize the discriminator, the discriminator would try to output 1 classifying it as real, for data coming from the target dataset $p_{\text{data}}(\boldsymbol{x})$ and output 0 classifying it as fake, for the data coming from the generator $G(\boldsymbol{z})$, which is the purpose of the discriminator.

The objective second step of the training process involves minimizing 2.16. Since this involves only tweaking the parameters of the Generator G, we can exclude the first term of the equation $\mathbb{E}_{\boldsymbol{x}\sim p_{\text{data}}(\boldsymbol{x})}[\log D(\boldsymbol{x})]$. Minimizing the second term would require $D(G(\boldsymbol{z})) \to 1$ for $\log(1 - D(G(\boldsymbol{z})))$ as $\lim_{x\to 0} f(x) = -\infty$. As D is supposed to output 1 only when the input data is from the target distribution $p_{\text{data}}(\boldsymbol{x})$, this step forces G to adjust its parameters such that the output is as close as possible to the target distribution. Playing this min-max game the resulting generator can ideally produce outputs consistent with the distribution of the target data, which cannot be distinguished by the discriminator, thus fooling it.

It is fairly straightforward to use adversarial learning in super-resolution:

$$\min_{\phi}\max_{\psi}\mathbb{E}_{I_{HR}\sim p_{\text{train}}(I_{HR})}\left[\log\mathcal{D}\left(I_{HR};\psi\right)\right]+\mathbb{E}_{I_{LR}\sim p_{G}(I_{LR})}\left[\log\left(1-\mathcal{D}\left(\mathcal{G}\left(I_{LR};\phi\right);\psi\right)\right]\right., \tag{2.16}$$

where the generator $\mathcal{G}$ is the SR model.The objective of the discriminator $\mathcal{D}$ in this context is to classify whether the input is generated or real. The first implementation of adversarial loss using cross entropy was introduced in SRGAN by Ledig et al. [13].

$$\mathcal{L}_{\text{gan\_G}}(\mathcal{G};\mathcal{D}) = -\log \mathcal{D}(\mathcal{G}\left(I_{LR};\phi\right);\psi), \tag{2.17}$$

where $\mathcal{L}_{\text{gan\_G}}$ is the generator adversarial loss, which takes in the output of the Generator $\mathcal{G}\left(I_{LR};\phi\right);\psi$, where $\mathcal{D}(\mathcal{G}\left(I_{LR};\phi\right);\psi)$ is the probability that the generated image is a natural image.

$$\mathcal{L}_{\text{gan\_D}}\left(\mathcal{G}, I_{HR};\mathcal{D}\right) = -\log \mathcal{D}\left(I_{HR};\psi\right) - \log\left(1-\mathcal{D}\left(\mathcal{G}\left(I_{LR};\phi\right);\psi\right)\right), \tag{2.18}$$

where $\mathcal{L}_{\text{gan\_D}}$ is the discriminator adversarial loss, which takes inputs from the target dataset I_{HR} and $\mathcal{G}\left(I_{LR};\phi\right);\psi$ to maximize (2.18). Enhancenet [70] also uses a similar loss.

The use of least square error based adversarial loss by Wang et al. [33] and Yuan at el. [73] demonstrated better stability in the training process and higher quality results [74], defined as:

$$\mathcal{L}_{\text{gan\_G}}(\mathcal{G};\mathcal{D}) = \left(\mathcal{D}(\mathcal{G}\left(I_{LR};\phi\right);\psi)\right)^2, \tag{2.19}$$

$$\mathcal{L}_{\text{gan\_D}}\left(\mathcal{G}, I_{HR};\mathcal{D}\right) = \left(\mathcal{D}\left(I_{HR};\psi\right)\right)^2 + \left(\left(1-\mathcal{D}\left(\mathcal{G}\left(I_{LR};\phi\right);\psi\right)\right)\right)^2, \tag{2.20}$$

Furthermore, Bulat et al. [75] demonstrated the use of a hinge based adversarial loss [76]:

$$\mathcal{L}_{\text{gan\_G}}(\mathcal{G};\mathcal{D}) = -\mathcal{D}(\mathcal{G}\left(I_{LR};\phi\right);\psi), \tag{2.21}$$

$$\mathcal{L}_{\text{gan\_D}}\left(\mathcal{G}, I_{HR};\mathcal{D}\right) = \min\left(0, \mathcal{D}\left(I_{HR};\psi\right)-1\right) + \min\left(0, \left(-\mathcal{D}\left(\mathcal{G}\left(I_{LR};\phi\right);\psi\right)\right)-1\right), \tag{2.22}$$

Work presented by Wang et al. [6] demonstrates the use of strongly argued concept of Relativistic average Discriminator [12], defined as:

$$\mathcal{L}_{\text{rgan\_G}}(I_{SR}, I_{HR}) = -\mathbb{E}_{I_{HR}\sim p_{\text{train}}}[log(\mathcal{D}_{\nabla}(I_{HR}, I_{SR})] - \mathbb{E}_{I_{SR}\sim p_{\text{SR}}}[log(1-\mathcal{D}_{\nabla}(I_{SR}, I_{HR})))], \tag{2.23}$$

$$\mathcal{L}_{\text{rgan\_D}}(I_{SR}, I_{HR}) = -\mathbb{E}_{I_{HR}\sim p_{\text{train}}}[log(1-\mathcal{D}_{\nabla}(I_{HR}, I_{SR})] - \mathbb{E}_{I_{SR}\sim p_{\text{SR}}}[log(\mathcal{D}_{\nabla}(I_{SR}, I_{HR})))], \tag{2.24}$$

where $I_{SR} = \mathcal{G}\left(I_{LR};\phi\right)$ and $\mathcal{D}_{\nabla}$ is the Relativistic average Discriminator. The standard discriminator in SRGAN [13] can be expressed as $\mathcal{D}(x) = \sigma(C(x))$, where σ stands for sigmoid and $C(x)$ denotes the non-transformed discriminator output [6]. Thus, $\mathcal{D}_{\nabla}(I_{HR}, I_{SR}) = \sigma(C(I_{HR} - \mathbb{E}_{I_{SR}}[C(I_{SR})]$, where $\mathbb{E}_{I_{SR}}[.]$ denotes the average for all super-resolved images (fake data). With this argument [6] claims to benefit from the gradients generated out of both the HR and SR image (real and fake data) in adversarial training. With the underlying intuition being, that the real images I_{HR} is relatively more realistic than a fake image I_{SR} [12].

In the work presented by Park et al. [77], argues that using a discriminator only at the pixel level causes the generator to add high-frequency noise. As a solution, an additional feature level discriminator is proposed to work on high-level features as discussed in "Content Loss" using pre-trained networks to extract better attributes of the target HR dataset.

2.3 Datasets

Table 2.1: Datasets for Image Super-Resolution

Dataset	Amount	Pixels	Format
BSDS300 [78]	300	154K	JPG
BSDS500 [79]	500	154K	JPG
Set5 [37]	5	113K	PNG
Set14 [80]	14	230K	PNG
DIV2K [81]	1000	2.7M	PNG
Urban100 [82]	100	774K	PNG
Manga109 [83]	109	966K	PNG

In the field of super-resolution, the data required for training is available in abundance, since it is very easy to procure images with variation compared to other fields of research. The available datasets vary a lot in terms of the number of images, quality, resolution variance. Some datasets are available in LR-HR pairs, for the others, it's usually a Bicubic Downsaple that is used to obtain the LR images. Even though this method of degradation is biased and might not reflect real-world scenarios, most of the work in literature use it. Though it is often coupled with analysis of other degradation methods. It has been demonstrated [10, 25], that the training dataset has a significant impact on the final results, and in most cases, a bigger dataset facilitates better results.

These state-of-the-art models discussed above are trained on various datasets and sometimes in a combination of various datasets [Table 2.1]. It lists some of the databases used in the super-resolution research describing number of images, the image format and an approximate number of pixels per image.

2.4 Evaluation

Having achieved the super-resolved image, it is critical to validate the results. This validation analysis can be done in one of two approaches. It can either be objectively or subjectively analyzed. Even though a subjective analysis would be ideal since in most scenarios, it's the human perceptual detail that would be required of the solution, it is very difficult and time-consuming to achieve such analysis. This is the primary reason for the extensive use of objective analysis as we need some form of comparison [84]. However, the evaluation of these two analyses usually isn't consistent with each other, as there have been differences measured in the evaluation results of objective and subjective studies [13, 85]. This section will further discuss some of the commonly used evaluation metrics, including both the above-mentioned analysis.

2.4.1 Peak signal-to-Noise Ratio

The peak signal-to-noise ratio (PSNR) is defined as the maximum possible power of a signal vs. the power of the noise that affects the representation.

$$PSNR = 10 \cdot \log_{10}\left(\frac{MAX_I^2}{MSE}\right), \tag{2.25}$$

where MAX_I^2 is the maximum possible power of a signal and MSE defines the noise of a signal. It is a quantitative way of measuring image reconstruction quality of a lossy transformation (Compression) on a scale of 0 to 100 dB, with higher values representing smaller Euclidean distance between two images. Likewise, in image super-resolution, PSNR is used to identify the reconstruction quality of the generated image. In the case of images, the maximum possible power of the signal is defined as $MAX_I = L = 255$, which is in general used for 8-bit images. Given a fixed L, PSNR is only dependant on pixel-wise MSE defined as:

$$\text{MSE} = \frac{1}{N}\sum_{i}^{N}(I_{SR}^{i} - I_{HR}^{i})^2, \tag{2.26}$$

Due to the focus on the difference between corresponding pixels and not the overall perceptual quality, PSNR is a poor representation of human visual perception [13, 70]. Despite being aware of such a scenario, PSNR is still widely used as an evaluation criterion for comparing state-of-the-art super-resolution models.

2.4.2 Structural Similarity

"*The main function of the human eyes is to extract structural information from the viewing field, and the human visual system is highly adapted for this purpose.*" [86]. Given this philosophy, the structural similarity index [85] was proposed to measure the structural similarity between images. It is another type of quantitative analysis based on three relatively independent comparisons; luminance, contrast, and structure. For any given image I with N pixels, the luminance component $\mathcal{C}_l(I_{HR}, I_{SR})$ is defined as:

$$\mathcal{C}_l(I_{HR}, I_{SR}) = \frac{2\mu_{I_{HR}}\mu_{I_{SR}} + C_1}{\mu_{I_{HR}}^2 + \mu_{I_{SR}}^2 + C_1}, \tag{2.27}$$

where $C_1 = (k_1 L)^2$ is a constant for avoiding instability, $k_1 << 1$ is a small constant and L is the maximum possible pixel value. μ_I is the mean of image intensity defined as:

$$\mu_I = \frac{1}{N}\sum_{i}^{N} I_i \tag{2.28}$$

Similarly the contrast component $\mathcal{C}_c(I_{HR}, I_{SR})$ is defined as:

$$\mathcal{C}_c(I_{HR}, I_{SR}) = \frac{2\sigma_{I_{HR}}\sigma_{I_{SR}} + C_2}{\sigma_{I_{HR}}^2 + \sigma_{I_{SR}}^2 + C_2}, \tag{2.29}$$

where $C_2 = (k_2 L)^2$ is a constant for avoiding instability, $k_2 << 1$ is a small constant and L is the maximum possible pixel value. σ_I is the standard deviation of image intensity defined as:

$$\sigma_I = \sqrt{\left(\frac{1}{N-1}\sum_i^N (I_i - \mu_I)^2\right)}, \tag{2.30}$$

The last component of comparison, the structure comparison function $\mathcal{C}_s(I_{HR}, I_{SR})$ is defined as:

$$\mathcal{C}_c(I_{HR}, I_{SR}) = \frac{\sigma_{I_{HR}I_{SR}} + C_3}{\sigma_{I_{HR}}\sigma_{I_{SR}} + C_3}, \tag{2.31}$$

where generally $C_3 = C_2/2$ and $\sigma_{I_{HR}I_{SR}}$ is the co-variance between I_{HR} and I_{SR} is defined as:

$$\sigma_{I_{HR}I_{SR}} = \frac{1}{N-1}\sum_i^N \left(I_{HR}^i - \mu_{I_{HR}}\right)\left(I_{SR}^i) - \mu_{I_{SR}}\right), \tag{2.32}$$

Finally, the SSIM is given by:

$$\text{SSIM}(I_{HR}, I_{SR}) = \left[\mathcal{C}_l(I_{HR}, I_{SR})\right]^\alpha \left[\mathcal{C}_c(I_{HR}, I_{SR})\right]^\beta \left[\mathcal{C}_s(I_{HR}, I_{SR})\right]^\gamma, \tag{2.33}$$

where α, β and γ are control parameters, which are generally set to 1. This brings SSIM to a specific form of:

$$\text{SSIM}(I_{HR}, I_{SR}) = \frac{(2\mu_{I_{HR}}\mu_{I_{SR}} + C_1)(\sigma_{I_{HR}I_{SR}} + C_2)}{\left(\mu_{I_{HR}}^2 + \mu_{I_{SR}}^2 + C_1\right)\left(\sigma_{I_{HR}}^2 + \sigma_{I_{SR}}^2 + C_2\right)}. \tag{2.34}$$

As stated earlier, SSIM evaluates the reconstruction quality from the perspective of the Human Visual System, it is a better fit for a perceptual loss [87, 88] and is used by most of the SR models. However, it can still be argued on how good a measure it is agains human perceptual system based on the work presented by Ledig et al. [13]. Furthermore, [89] demonstrates similarity between PSNR and SSIM by showing how SSIM is just less sensitive to additive Gaussian noise and PSNR can be approximated given SSIM.

2.4.3 Mean Opinion Score

As discussed above, the quantitative approach towards image quality assessment has some limitations in comparison with the image's true perceptual quality. For this reason, a lot of work has demonstrated the use of human evaluation [90, 13, 31].Even though a qualitative approach has its limitations in execution, ideally, it is probably the best option as of now to identify true perceptual quality. When performing an opinion score, the human raters are usually asked to rate an image with a score range of 1(poor) to 5(excellent) [13]. With the user inputs, MOS is calculated as the mean over the user ratings. MOS has it's own sets of limitations, as it involves a lot of different human perspectives, which can be biased and reviews might vary among different users. But statistically given a large sample size should theoretically solve this issue. In literature, some SR models have demonstrated to perform poorly on the objective-based evaluations like PSNR and SSIM, but have faired highly with subjective evaluation like MOS. With this, it is safe to assume that MOS testing is the best option for measuring perceptual quality [13, 70, 90, 31].

Chapter 3

Network Architecture

3.1 Overview

The overall aim of this research is to investigate a suitable network architecture, that can perform single image super-resolution at real-time interactive rates. This would require, for any given image, upscaling it by the desired scale factor ideally under 41.66 ms to achieve 24 fps, 33.33 ms to achieve 30 fps and 16.66 ms for 60 fps. Keeping these constrains in mind this work analyses all the network architectures discussed in related work. This is coupled with extensive experiments on most of them to find the best network components that maximize contribution to better image reconstruction with minimum computational cost. Along with that, this work analyses some design choices and techniques very closely which best fit to solve the research problem.

3.2 Experimental Design

As discussed in the literature review, with advances in this field of research, the reconstructed images are incrementally getting better, but at the cost of very deep network, which might be impractical in terms of real time execution even with our modern GPUs. Thus, in order to design a good architecture to reconstruct visually pleasing images in real-time, it was import to focus on smaller networks [10, 24, 25]. Even though SRCNN [10] is a very fast solution today due to the advances in hardware, it wasn't the same earlier, which gave rise to the scope for real-time super-resolution networks like FSRCNN [24] and ESPCN [25]. The network architecture proposed in this work follows similar design choice of being minimal in nature to perform in real-time at the same time adopting newer design principals. In order to design the final network architecture several experiments were conducted to deduce some of the best design principals.

3.2.1 Baseline Benchmark

The first implementation of Deep Learning based super-resolution architecture SRCNN [10] is used as the baseline for benchmarking. SRCNN compares and demonstrates how sparse-coding-based super-resolution models [37, 91], can be seen as convolutional neural networks. It is accordingly designed to be a simple three layered design. The first layer is called "*Patch extraction and representation*", the second layer is "*Non-linear mapping*" and the last layer is "*Reconstruction*" layer.

The network architecture proposed outperformed the state-of-the-art traditional methods [10, 2]. SRCNN is a fairly simple formulation of a deep neural network architecture using the pre-upsampling method, discussed in (Background). It demonstrates the ability of convolutional neural networks to have representational learning in an end-end fashion. Analyzing the network structure along with the rest of the literature highlights some important questions:

- Is it possible for the network to directly learn low-resolution to high-resolution mapping?
- Does the network learn better representation with increase in depth and width?
- Does the use of Generative Adversarial Network help with synthesizing high-frequency information?
- How can the change in loss function affect perceptual quality?

These questions have been extensively covered in literature and further experiments demonstrate the properties of each of the networks answering these questions, thus enabling to deduce and formulate a more efficient network architecture suitable for the research question.

3.2.2 Efficient Upsampling

To address the first question, Dong et al. [24] proposed the use of Deconvolutional Network [41] and Shi et al. [25] introduced the use of Pixel Shuffle network to upscale the image at the final step of the super-resolution process. As Shi et al. [11] further elaborates on the comparison on Deconvolutional Network vs Sub-Pixel Convolutional Networks, stating that the Sub-Pixel network is the same as the deconvolutional network in terms of speed, but has larger number of parameters, giving it greater representational power.

To conclude the arguments presented in [11], several experiments have been conducted on the above mentioned baseline network. Firstly, we train the baseline SRCNN on 48 × 48 random crop of DIV2K [81] with validation on Set5 [37] and time to super-resolve a single 1920 × 1080 image by 2× scaling factor to achieve 4K image for comparison. With the same configuration of SRCNN, unlike the one in [25] for consistency, a Sub-Pixel convolutional network is added at the end as presented in [25]. This is followed by replacing the Sub-Pixel network with a deconvolutional network [41] with the optimum parameters as observed in DBPN [1]. With this set, the PSNR, SSIM and time per frame in seconds is recorded to evaluate the claims made in [11]. As it can be seen from [Table 3.1]:

Network	PSNR	SSIM	Param	Time
SRCNN	35.2727	0.9422	$20,099$	$0.002s$
SRCNN˙Dconv	34.8920	0.9378	$28,607$	$0.002s$
SRCNN˙PixShufle	35.1169	0.9478	$46,796$	$0.002s$

Table 3.1: Comparison

It can be said that the claims made in [11] by Shi et al. stands valid, as it argued that despite having higher parameters the network complexity remains the same. Even though the term "*same speed*" wasn't clearly defined, with this experiment we can confirm the speed is certainly the same but with higher parameters thus giving it better "*representation power*". We can conclude that the use of Pixel-Shuffle network is a good design choice over the deconvolutional network. Note, Both the pixel-shuffle based and deconvolution based network perform poorly in comparison with their originals propositions FSRCNN [24] and ESPCN [25], as the network architecture preceding the final upscale layers is exactly the same as SRCNN [10], unlike in their original implementations.

Argument 1 *Use of Sub-Pixel Convolution is better than Deconvolutional Network for Image Super-Resolution.*

3.2.3 Deeper and Wider Networks

"*A feedforward neural network is a composition of layers of computational units which defines a function*" $F : \mathbb{R}^c \rightarrow \mathbb{R}^d$ [92]. This is a theoritical work shwocasing that the solution space of a Deep Neural Network can be expanded by increasing its width or depth [2]. Ever since SRCNN demonstrated its potential, many Deep learning based applications have been demonstrated to use deeper networks to further increase representation power. VDSR [21] is to the best of knowledge the first very deep super-resolution architecture. It introduced the use of a 20 layer VGG network [72] along with Global Residual Learning. It also uses the pre-upsample method like SRCNN, thus making it computationally slightly inefficient.

Subsequently Lee et al. proposed EDSR [30], which makes use of the Local Residual Learning, where the batch Normalization layer is removed. It is argued that Batch Normalization is not suitable for super-resolution as it removes the range exibility of the features. Along with increasing the depth of the residual network [3], EDSR also widens the network by using higher filters. To accommodate for issues regarding wide ResNet, residual scaling technique is used from [93].

Along with ResNet, DenseNet [35] is also a very effective choice of skip connection based network. As described in [94], " *ResNet enables feature re-usage while DenseNet enables new features exploration which are both important for learning good representations*". Based on which, SRDenseNet [26], was proposed, demonstrating effectiveness in performance using the post-upsample method. Based on the same concept, Haris et al. proposed the Deep Back-Projection Network [1]. The primary motivation behind this work is, that a feedback mechanism can significantly boost the performance of mapping from low-resolution to high-resolution in comparison to a feed-forward model. To demonstrate this, the network architecture consists of a series of up an down sample layers which are densely connected. The main buildfing block of the DBPN network is the projection unit, which is trained to map the low-resolution image to high-resolution using the up-projection and the other way using the down-projection block.

The up-projection block is defined as: [1]

$$\mathbf{H}_0^t = (\mathbf{L}^{t-1} * \mathrm{d}_0^t) \uparrow_s, \tag{3.1}$$

$$\mathbf{L}_0^t = (\mathbf{H}_0^t * \mathrm{d}_1^t) \downarrow_s, \tag{3.2}$$

$$\mathbf{e}_t^l = \mathbf{L}_0^t - \mathbf{L}^{t-1}, \tag{3.3}$$

$$\mathbf{H}_1^t = (\mathbf{e}_t^l * \mathrm{d}_2^t) \uparrow_s, \tag{3.4}$$

$$\mathbf{H}^t = (\mathbf{H}_0^t + \mathbf{H}_1^t). \tag{3.5}$$

where $*$ is the convolution operator, $\uparrow_s$ is the up-sampling operator and $\downarrow_s$ is the down-sampling operator. d_i^t is the deconvolutional layer at stage t with scaling factor s.The projection unit takes the low-resolution feature map $\mathbf{L}^{t-1}$ and projects it to the first high-resolution map $\mathbf{H}_0^t$, which is the scale up phase. Subsequently it back projects this intermediate high-resolution feature map to a low resolution feature map $\mathbf{L}_0^t$, which is the scale down phase. Following this, it takes a residual between low-resolution feature map $\mathbf{L}^{t-1}$ and the back projected low-resolution feature map $\mathbf{L}_0^t$. This is followed by another up-scale step, which upscales the residual feature map $\mathbf{e}_t^l$, forming the second high-resolution feature map $\mathbf{H}_1^t$. Finally the high-resolution feature map from the up-projection block $\mathbf{H}^t$ is obtained by summing the first and second intermediate high-resolution feature maps $\mathbf{H}_0^t$ and $\mathbf{H}_1^t$.

The down-projection block is defined similarly, which is doing the inverse of the up-projection block. It maps the high-resolution feature map input $\mathbf{H}^t$ to the low-resolution feature map $\mathbf{L}^t$ using c_i^t convolutions instead of deconvolutions. It forms the input to the up-projection block in the subsequent stage $t + 1$.

The down-projection block is defined as: [1]

$$\mathbf{L}_0^t = (\mathbf{H}^t * \mathrm{c}_0^t) \downarrow_s, \tag{3.6}$$

$$\mathbf{H}_0^t = (\mathbf{L}_0^t * \mathrm{c}_1^t) \uparrow_s, \tag{3.7}$$

$$\mathbf{e}_t^h = \mathbf{H}_0^t - \mathbf{H}^t, \tag{3.8}$$

$$\mathbf{L}_1^t = (\mathbf{e}_t^h * \mathrm{c}_2^t) \downarrow_s, \tag{3.9}$$

$$\mathbf{L}^t = (\mathbf{L}_0^t + \mathbf{L}_1^t). \tag{3.10}$$

Both up and down projection units are illustrated in [Figure 3.1]

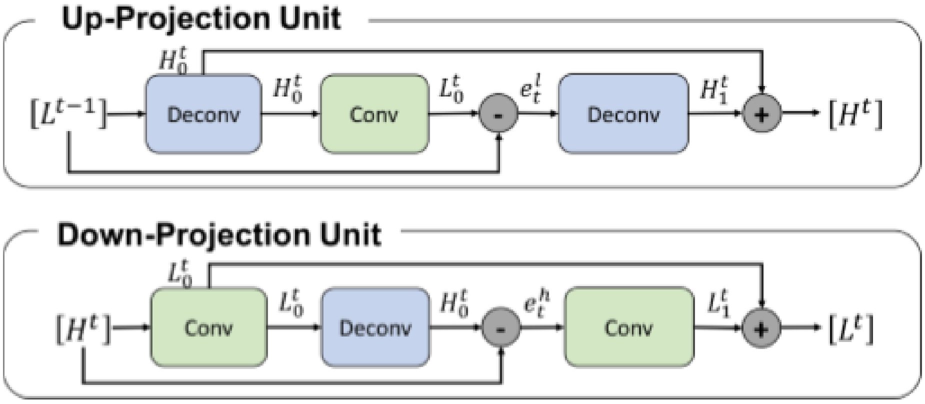

Figure 3.1: up and down projection unit in DBPN [1]

DBPN, further demonstrates the use of dense connection as an improvement, proposing the Dense-DBPN (D-DBPN) As shown in [30] and discussed above, batch normalization isn't beneficial to super-resolution, DBPN, makes use of 1 × 1 convolution layers as feature pooling before using the back-projection unit instead. An illustration of the dense up and down projection unit can be seen in [Figure 3.2]

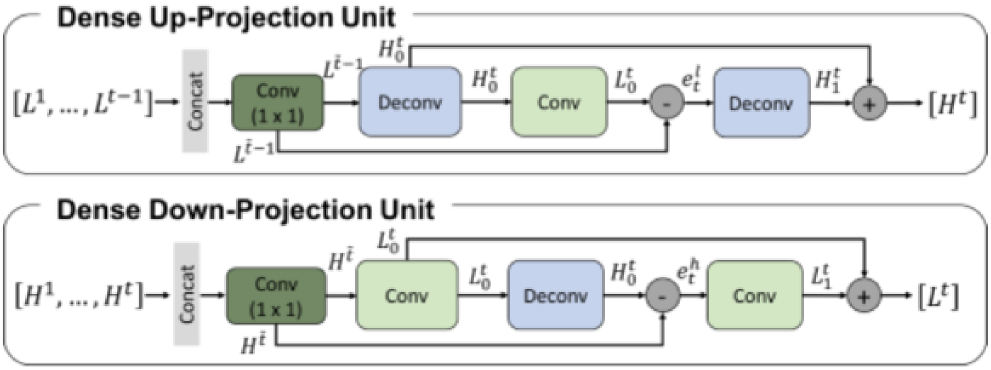

Figure 3.2: Dense up and down projection unit in DBPN [1]

The network architecture for D-DBPN is divided into three parts: "*Initial feature extraction*", "*Back-projection stages*" and "*reconstruction*". On comparing this with the basline model, it can be observed that DBPn also follows a similar architectural pattern. It uses the Back-projection block as the "*Non-linear mapping*" [10]. Furthermore, Haris et al. [1] states that, to their best knowledge a feedback network has never been implemented for super-resolution till 2018. DBPN demonstrates really strong and promising results in comparison with other state-of-the-art.

With this said, the gap was filled in this domain by Li et al. [5] introducing super-resolution feedback network (SRFBN) for image super-resolution. Li et al. [5] argues that with the increase in depth of a network the parameters increase and such large capacity networks can suffer from the overfitting problem. The recurrent structure is often used to reduce the number of parameters in a network. SRFBN introduces the use of a feedback network [36], "*in order to rene low-level information using high-level one through feedback connections*". The feedback mechanism proposed helps the network to generate better images by correcting the generated images in previous states. Even though DBPN [1] and DSRC [27] proposed feedback mechanisms, these steps were still carried out in a feed-forward fashion, unlike in SRFBN. The network architecture in SRFBN can also be shown to be similar to our baseline, as it uses a "*LR feature extraction block*" (LRFB) similar to the first layer of our baseline. It has some difference in the way this layer is constructed. Instead of using a single convolutional layer, it uses two layers a feature extraction layer $Conv(3 \times 3)$ and a feature reduction layer $Conv(1 \times 1)$. The "*Feedback block*" is used as the "*Non-linear mapping*" from our baseline following it with a post-upsampling method by using a reconstruction block which is a deconvolutional network. It also employs the use of Global Residual Learning as discussed in literature review, as the network learns the residual between the bicubic upsample and the target image.

The Feedback network is constructed out of an iterative up and down-sampling blocks which are densely connected. As it can be seen in [Figure 3.3], the Feedback block takes at time step t input F^{t-1}_{out}, which is the output of the feedback block from previous time step to correct the low-level representation F^{t}_{out} subsequently passing higher-level representation F^{t}_{out} to the next iteration $t+1$ and finally to the reconstruction block. The Feedback block can be defined as [5]:

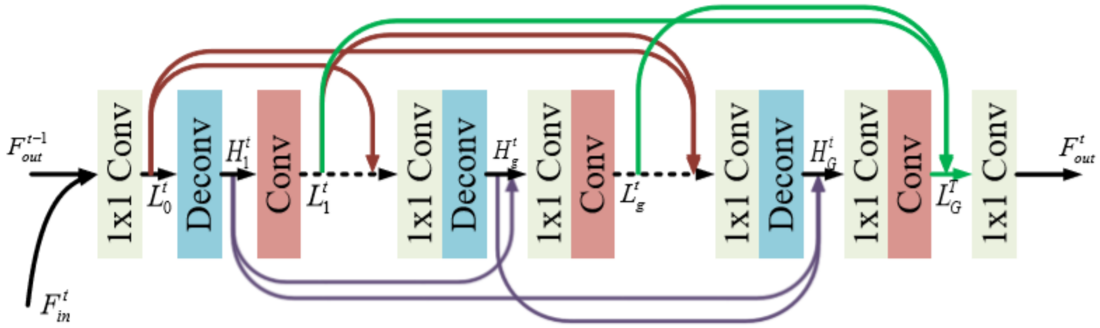

Figure 3.3: Feedback Block in SRFBN [5]

$$L_0^t = C_{in}\left(\left[F_{out}^{t-1}, F_{in}^t\right]\right) \tag{3.11}$$

$$H_g^t = C_g^{\uparrow}\left(\left[L_0^t, L_1^t, \ldots, L_{g-1}^t\right]\right) \tag{3.12}$$

$$L_g^t = C_g^{\downarrow}\left(\left[H_1^t, H_2^t, \ldots, H_g^t\right]\right) \tag{3.13}$$

$$F_{out}^t = C_{out}\left(\left[L_1^t, L_2^t, \ldots, L_G^t\right]\right) \tag{3.14}$$

where, C_{in} and c_{out} are compression layers defined as $Conv(1 \times 1)$, used for dimensional reduction and [.] is referred to concatenation. $C_g^{\uparrow}$ and $C_g^{\downarrow}$ are up-sampling and down-sampling group, where the network consists of G projection groups sequentially with dense skip connections. In the first step, the network takes in F_{in}^t and F_{out}^{t-1} and uses the initial compression unit to refine input features, resulting refined feature L_0^t Subsequently by concatenating all previously generated low-resolution feature maps $(L_0^t, \ldots, L_{g-1}^t)$ in the Feedback block are taken as the input by the g^{th} up-sampling group $C_g^{\uparrow}$ producing high-resolution feature map H_g^t. Likewise all generated high-resolution feature maps $(H_0^t, \ldots, H_g^t)$ are back-projected using the down-sampling group $C_g^{\downarrow}$ to form the next low-resolution feature map L_g^t. Finally all low-resolution feature maps generated in the Feedback block are concatenated and passed to C_{out} compression unit to produce a refined low-resolution feature map as the Feedback blocks output. The argument made in this work, that the use of a recurrent neural network to improve reconstruction quality by reducing network parameter is demonstrated very well. However, the use of a recurrent structure hinders the inference performance significantly.

3.2.4 Proposed Baseline Network

The work presented in state-of-the-art discussed above demonstrates better performance with increase in network width and depth. The comparison of the discussed networks will be later discussed in the [Results Section]. Based on observations made in these deep networks, some experiments were performed to conclude an efficient shallow baseline. The purpose of concluding a shallow baseline, is to demonstrate its potential for real-time performance and to subsequently build on top of the same design principal to go deeper. The proposed baseline also adopts a three stage architecture, with a Feature Extraction layer, Non-Linear mapping and final Reconstruction layer. Based on observation from the state-of-the-art, and [Argument 1] the use of post-upsample method was the most appropriate design choice. Subsequently Global Residual Learning is employed.

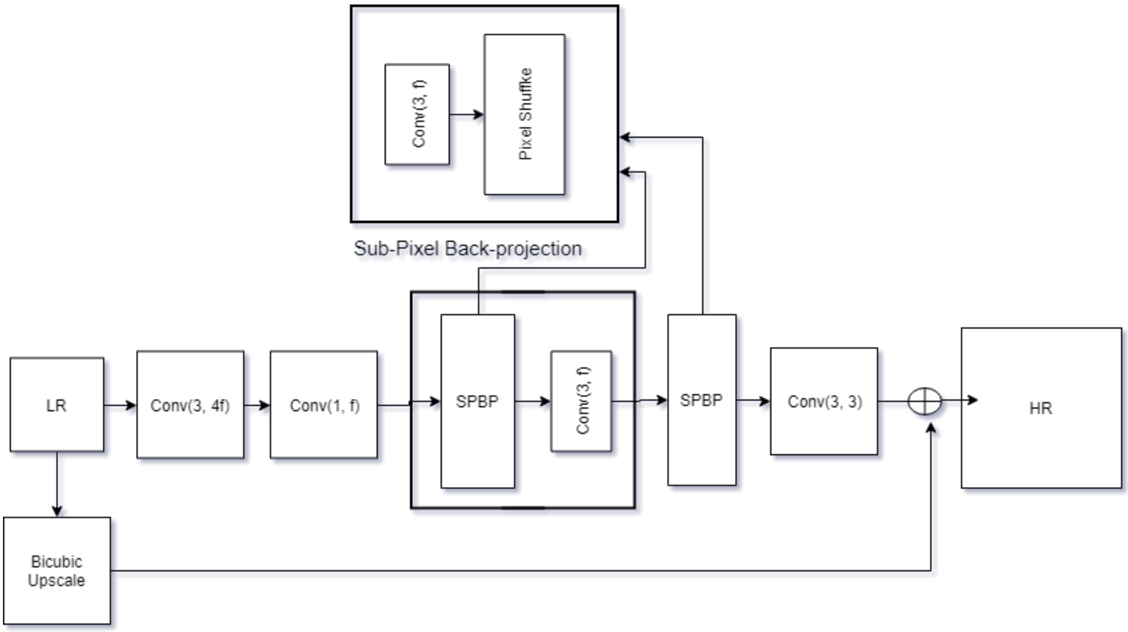

Figure 3.4: Proposed Baseline Architecture

The Global Residual Learning used in the network which bypasses the sub-network by forwarding an upsampled image at the end which is added to the output of Reconstruction layer. The proposed network architecture is defined as following, where a convolutional layer is denoted as $Conv(k, n)$ and Deconvolutional layer as $Deconv(k, n)$, where k is the filter size and $n$is the number of filters.

Feature Extraction

The Feature Extraction block is defined as:

$$\mathcal{F}_{in}^{0} = \mathcal{C}_0^{FE}(I_{LR}), \tag{3.15}$$

$$\mathcal{F}_{in}^{1} = \mathcal{C}_1^{FE}(\mathcal{F}_{in}^{0}), \tag{3.16}$$

where $\mathcal{C}_0^{FE} = Conv(3, 4f)$ and $\mathcal{C}_0^{FE} = Conv(1, f)$, where f is the base number of filters. The low-level representation F_{in}^{0} is obtained from the low-resolution image I_{LR} and the refined feature F_{in}^{1} is obtained by F_{in}^{0}.

Non-Linear Mapping

The Non-Linear Mapping layer is a Back-projection block, similar to that in SRFBN [5] which is a simplified version of DBPN [1]. Instead of using a Deconvolution layer, based on [Argument 1], a Sub-Pixel Convolution based up-projection layer is implemented. The Non-Linear Mapping layer is defined as:

$$\mathcal{H}_0 = \mathcal{PS}(\mathcal{C}_0^{BP}(\mathcal{F}_{in}^{1}) \uparrow_s, \tag{3.17}$$

$$\mathcal{L}_0 = \mathcal{C}_1^{BP}(\mathcal{H}_0) \downarrow_s, \tag{3.18}$$

where $\mathcal{H}_0$ is the high-resolution feature map and $\mathcal{L}_0$ is the low-resolution feature map. $\uparrow_s$, $\downarrow_s$ represent upsample and downsample operation respectively with scale factor$s$. $\mathcal{C}_0^{BP}$ represents $Conv(3, fs^2)$, f is the number of filter, s is the scale factor and $\mathcal{PS}$ is the Pixel-Shuffle Layer. The back-projection block takes $\mathcal{F}_{in}^{1}$ as input and produces a high-resolution feature map $\mathcal{H}_o$, which is back projected to a low-resolution feature map $\mathcal{L}_o$‘. This low-resolution feature map is the passed to the Reconstruction layer. This non-linear mapping is called the Sub-Pixel Back-Projection Block.

Reconstruction

As discussed above, the Reconstruction layer is a Sub-pixel convolutional layer [Argument 1] which upscales the low-resolution feature map to the desired scaling factor.

The Reconstruction layer is defined as:

$$\mathcal{I}_0^{Res} = \mathcal{PS}(\mathcal{C}_0^R(\mathcal{L}_0) \uparrow_s, \tag{3.19}$$

$$\mathcal{I}_1^{Res} = \mathcal{C}_1^R(\mathcal{I}_0^{Res}), \tag{3.20}$$

$$\mathcal{I}_{SR} = \mathcal{I}_1^{Res} + f_{UP}(\mathcal{I}_{LR}), \tag{3.21}$$

where $\mathcal{I}_0^{Res}$ is the residual upscale using Sub-Pixel convolution $\mathcal{PS}(\mathcal{C}_0^R(\mathcal{L}_0))$ with input $\mathcal{L}_0$ from the Non-Linear Mapping layer. $\mathcal{I}_1^{Res}$ is the refined residual high-resolution feature map derived from $\mathcal{C}_1^R()$, which is a $Conv(3, f_{out})$ where, $f_{out} = 3$ is the output feature map "RGB". Finally the super-resolved image is constructed by adding the high-resolution refined feature map with $f_{UP}(.)$, which is linear upsample of the low-resolution image. Since the low-resolution image contains abundant low-frequency information [4], this allows the network to bypass the low-resolution information and focus only on the residual component from the high-resolution image. Experimental results to conclude this design choice based on the same evaluation parameters used for [Argument 1] are shown in [Table 3.2]

Network	PSNR	SSIM	Param	Time
SRCNN	35.2727	0.9422	$20,099$	$0.002s$
Baseline˙Dconv	36.4480	0.9522	$17,733$	$0.002s$
Baseline˙SP	36.5662	0.9528	$24,421$	$0.002s$

Table 3.2: New Baseline Comparison

where Baseline-Dconv is the above discussed baseline with a deconvolutional upsampling and Baseline-SP is the baseline with sub-pixel convolutional upsameple. Given these results, we can conclude that the proposed baseline is superior to SRCNN [10], and two modified version of SRCNN to establish [Argument 1], discussed in [3.2.2 "*Efficient Upsampling*"]. Further experiments will be based on the newly defined Baseline.

Argument 2 *Use of Sub-Pixel Convolution is better than Deconvolutional Network for shallow Back-projection unit.*

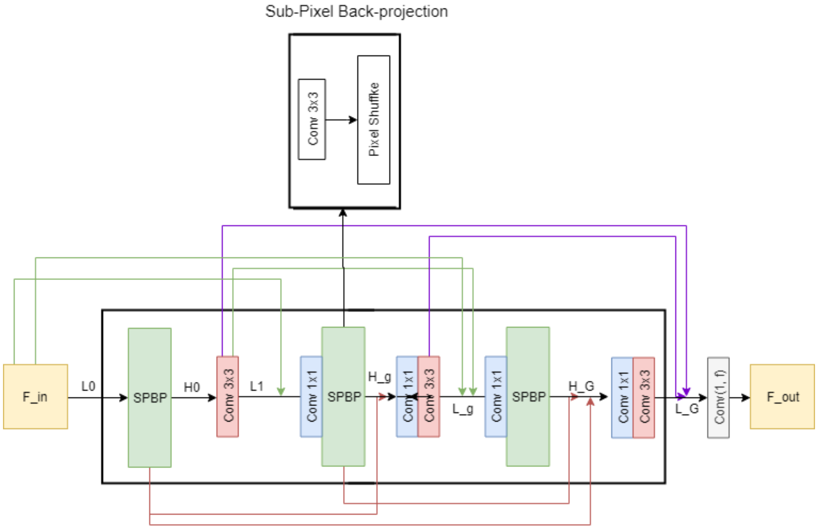

Figure 3.5: Sub-Pixel Back-Projection Block

Sub-Pixel Back-Projection Block

The design of Sub-Pixel Back-Projection (SPBP) is shown in [Figure 3.5]. It takes the input F_{in} which is denoted as L_0 in the illustration. The construct of the Back-Projection unit is very similar to that of SRFBN [5]. It does not use the initial compression convolution and the Deconvolution in SRFBN. The Sub-Pixel Back-Projection block can thus be formulated as:

$$L_0 = F_{in} \tag{3.22}$$

$$H_g = \mathcal{PS}(\mathcal{C}_g^{BP}\left([L_0, L_1, \ldots, L_{g-1}]\right)) \uparrow_s, \tag{3.23}$$

$$L_g = \mathcal{C}_g^{BP}\left([H_1^t, H_2^t, \ldots, H_g^t]\right) \downarrow_s, \tag{3.24}$$

$$F_{out} = C_{out}\left([L_1, L_2, \ldots, L_G]\right) \tag{3.25}$$

3.2.5 Adversarial Training

In this section the use of Generative Adversarial Networks [14] in super-resolution is extensively discussed. The purpose of using GANs was to achieve visually pleasing results and not minimizing the pixel distance to achieve higher PSNR values. For this purpose Ledig et al. proposed SRGAN [13] which uses an adversarial objective function that encourages the super-resolved image to be similar to that of the target images. The primary contribution made by SRGAN is the use of a perceptual loss, which makes use of a content loss and an adversarial loss (3.26).

$$I^{SR} = \underbrace{l_{\mathrm{X}}^{SR}}_{\text{content loss}} + \underbrace{10^{-3} l_{Gen}^{SR}}_{\text{adversarial loss}} \tag{3.26}$$

where l_{X}^{SR} is the content loss which uses a MSE loss to find distance pixel-wise I_{MSE}^{SR} (3.27) and distance metric for perceptually similarity over high-level feature maps drawn from a pretrained VGG19 network .

$$I_{MSE}^{SR} = \frac{1}{HW} \sum_{x}^{W} \sum_{y}^{H} \left(\mathcal{G}\left(I^{LR}; \phi\right)_{x,y} - I_{x,y}^{HR} \right)^2 , \tag{3.27}$$

$$I_{VGG/\ell}^{SR} = \frac{1}{W_\ell H_\ell} \sum_{x}^{W} \sum_{y}^{H} \left(\Psi_{x,y}^{(\ell)}(\mathcal{G}\left(I^{LR}; \phi\right)) - \Psi_{x,y}^{(\ell)}(I^{HR}) \right)^2 , \tag{3.28}$$

where $\mathcal{G}\left(I^{LR}; \phi\right)_{x,y}$ is the generated image, $I_{x,y}^{HR}$ is the target image and $\Psi_{x,y}^{(\ell)}$ is the output from tehe pretrained network at layer ℓ. Thus $l_{\mathrm{X}}^{SR} = I_{MSE}^{SR} + I_{VGG/\ell}^{SR}$. Subsequently the adversarial loss l_{Gen}^{SR} is defined as:

$$I_{Gen}^{SR} = -\log \mathcal{D}\left(\mathcal{G}\left(I^{LR}; \phi\right); \psi\right), \tag{3.29}$$

where $\mathcal{D}\left(\mathcal{G}\left(I^{LR}; \phi\right); \psi\right)$ is the probability of the reconstructed image being a natural target image.

Bulding up the work of Ledig et al. [13], Wang et al. proposed Enhanced - SRGAN (ESRGAN) [6]. It has its primary contribution in two fold in the modification of the SRGAN generator network, by firstly removing the batch normalization layer and replacing the original basic block with a Residual-in-Residual Dense Block (RRDB) [Figure 3.6] [Figure 3.7] .

Along with employing a novel RRDB block in the generator, ESRGAN also introduces the concept of a Relativistic average Discriminator [12]. As discussed in related work (Adversarial Loss), the discriminator $\mathcal{D}$ in ESRGAN is different in comparison to the standard GAN used in SRGAN. The discriminator in SRGAN estimates the

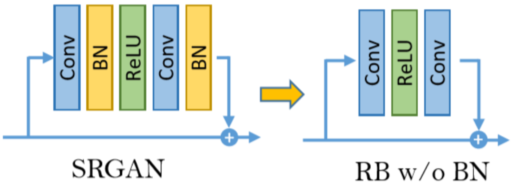

Figure 3.6: Batch Normalization removed in SRGAN [6]

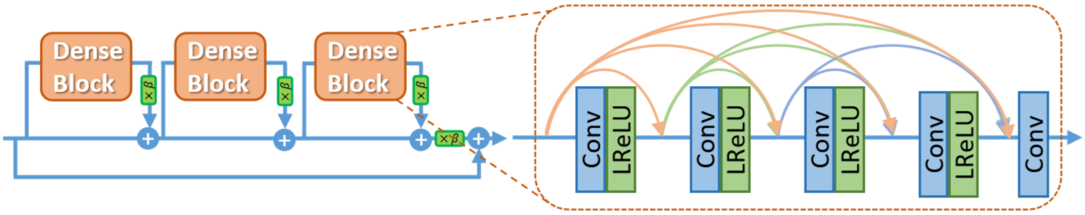

Figure 3.7: RRDB block used in ESRGAN [6]

probability of an input image being real, where as the relativistic discriminator estimates the probability that a real image is relatively more realistic than a fake image [6]. This can be seen in [Figure 3.8] : Similar to the relativistic discriminator's intuition,

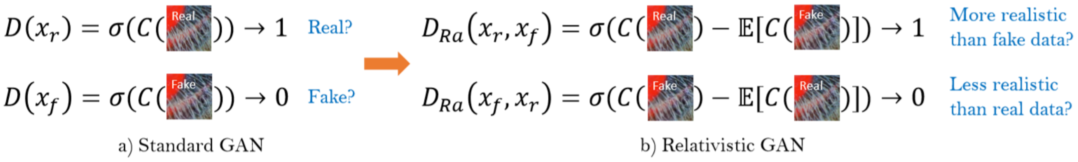
$D(x_r) = \sigma(C($ Real $)) \rightarrow 1$
Real?
$D(x_f) = \sigma(C($ Fake $)) \rightarrow 0$
Fake?
a) Standard GAN
$D_{Ra}(x_r, x_f) = \sigma(C($ Real $) - \mathbb{E}[C($ Fake $)]) \rightarrow 1$
More realistic than fake data?
$D_{Ra}(x_f, x_r) = \sigma(C($ Fake $) - \mathbb{E}[C($ Real $)]) \rightarrow 0$
Less realistic than real data?
b) Relativistic GAN

Figure 3.8: SGAN vs RGAN [6]

ESRGAN specifically incorporates the Relativistic average Discriminator (RaD) [12]. from [Figure 3.8] we can describe a Standard GAN as $\mathcal{D}(x) = \sigma(C(x)$, where σ is the sigmoid activation and $C(x)$ is the non-transformed discriminator output [12]. Thus RaD is formulated as $\mathcal{D}_{Ra}(x_r, x_f) = \sigma(C(x_r) - \mathbb{E}_{x_r}[C(x_f)])$, where $\mathbb{E}_{x_f}[.]$ is an average operation for all fake data. The use of RaGAN has demonstrated to learn sharper edges and detailed textures [6].

To further improve perceptual quality Park et al. [77] proposed super-resolution using Feature Discriminator (SRFeat). In this work, it is argued that the GAN-based networks used to synthesize high-frequency textures tent to add less meaningful high-frequency noise to the generated image, which is irrelevant to the input image. It employs a Standard GAN [14] to train the generator network the loss function minimized is:

$$L_g = L_p + \lambda(L_a^i + L_a^f) \tag{3.30}$$

where L_p is a perceptual similarity loss which directs the super-resolved images to look similar to the ground truth target images, similar to $I^{SR}_{VGG/\ell}$ Eq. (3.28) in SRGAN. SRFeat uses this perceptual similarity loss measuring difference in the feature domain instead of measuring it pixel-wise. L_a^i is the adversarial image/generator loss similar to I^{SR}_{Gen} Eq. (3.29). The additional component in comparison with SRGAN is the use of L_a^f which is the feature adversarial loss. The feature adversarial loss is defined as:

$$L_a^f = -\log \mathcal{D}_f \left(\Psi^{(\ell)}(\mathcal{G}\left(I^{LR}; \phi\right)); \psi \right), \tag{3.31}$$

where, $\Psi^{(\ell)}$ is a feature extraction at layer ℓ and $\mathcal{D}_f$ is the feature discriminator, which gives the probability of feature map being from the distribution of the target high-resolution feature maps. "*As features correspond to abstracted image structures, we can encourage the generator to produce realistic structural high-frequency rather than noisy artifacts*" [77]. Further the discriminator adversarial loss L_d^f is defined as:

$$L_d^f = -\log(\mathcal{D}_f \left(\Psi^{(\ell)}(I^{HR})\right); \psi) - \log(1 - \mathcal{D}_f \left(\Psi^{(\ell)}(\mathcal{G}\left(I^{LR}; \phi\right)); \psi\right)), \tag{3.32}$$

This work successfully demonstrates the use of two discriminator networks for working with the feature domain and image domain which leads to better perceptual quality.

3.2.6 Proposed Adversarial Learning

The demonstrations made in the works discussed above highlight the importance of adversarial training in image super-resolution. Since we are trying to predict missing information, learning the probability distribution of the training set seems to be an ideal solution coupled with a perceptually based loss. Based on the observations made in literature and above discussed training methods, several experiments were conducted to derive the most suitable design for adversarial learning. The proposed adversarial network architecture uses a Relativistic average Discriminator [12] which is used in the image domain and feature domain to produce better structural high-frequency information [77].

To train the generator network, the loss function $\mathcal{L}_{gen}^{SR}$ is minimized:

$$\mathcal{L}_{gen}^{SR} = \mathcal{L}_{c}^{SR} + \lambda \mathcal{L}_{a}^{SR} \tag{3.33}$$

$$\mathcal{L}_{c}^{SR} = (\alpha \mathcal{L}_{c}^{SR_i} + \gamma \mathcal{L}_{c}^{SR_{f_\ell}}) \tag{3.34}$$

$$\mathcal{L}_{a}^{SR} = \mathcal{L}_{ga}^{SR_i} + \mathcal{L}_{ga}^{SR_{f_\ell}} \tag{3.35}$$

where, $\mathcal{L}_{c}^{SR}$ is the content loss and $\mathcal{L}_{a}^{SR}$ is the generator adversarial loss with weight λ. $\mathcal{L}_{c}^{SR_i}$ is the generator content loss in the image domain, $\mathcal{L}_{c}^{SR_{f_\ell}}$ is the generator content loss in the feature domain with α and γ as contribution weights for respective domain losses. $\mathcal{L}_{ga}^{SR_i}$ is the generator adversarial loss in the image domain, $\mathcal{L}_{ga}^{SR_{f_\ell}}$ is the generator adversarial loss in the feature domain.

Content Loss

The content wise loss is comprised of two components, the first being the pixel-wise loss in the image domain $\mathcal{L}_{c}^{SR_i}$, which is a L1 loss defined as:

$$\mathcal{L}_{c}^{SR_i} = \frac{1}{HWC} \sum_{x}^{W} \sum_{y}^{H} \sum_{i}^{C} \left\| \mathcal{G}\left(I_{LR}; \phi\right)^{x,y,i} - I_{HR}^{x,y,i} \right\|, \tag{3.36}$$

where H, W and C denote the dimensions of the target $I_{HR}^{x,y,i}$ and generated image $\mathcal{G}\left(I_{LR}; \phi\right)$. Similarly the second part of the content loss, which is in the feature domain $\mathcal{L}_{c}^{SR_{f_\ell}}$ also uses a L1 loss, defined as:

$$\mathcal{L}_{c}^{SR_{f_\ell}} = \frac{1}{HWC} \sum_{x}^{W} \sum_{y}^{H} \sum_{i}^{C} \left\| \Psi_{x,y,i}^{(\ell)}(\mathcal{G}\left(I^{LR}; \phi\right)) - \Psi_{x,y,i}^{(\ell)}(I^{HR}) \right\|, \tag{3.37}$$

where H, W and C denote the dimensions of the ℓ^{th} feature map $\Psi^{(\ell)}$. Similar to commonly used VGG network, the network used in this experiment is a VGG-19 [72].

Adversarial Loss

The adversarial loss can be classified in two segments. The first being the generator adversarial loss and second, discriminator adversarial loss. The loss functions in Standard GAN [14] with respect to image super-resolution can be expressed as:

$$\mathcal{L}_{\mathcal{D}} = \mathbb{E}_{I_{HR} \sim \mathbb{P}} \left[log \left(\mathcal{D} \left(I_{HR} \right) \right) \right] + \mathbb{E}_{I_{LR} \sim \mathbb{Q}} \left[log \left(1 - \mathcal{D} \left(\mathcal{G} \left(I_{LR} \right) \right) \right) \right] \tag{3.38}$$

$$\mathcal{L}_{\mathcal{G}} = \mathbb{E}_{I_{HR} \sim \mathbb{P}} \left[log \left(1 - \mathcal{D} \left(I_{HR} \right) \right) \right] + \mathbb{E}_{I_{LR} \sim \mathbb{Q}} \left[log \left(\mathcal{D} \left(\mathcal{G} \left(I_{LR} \right) \right) \right) \right] \tag{3.39}$$

where $\mathcal{L}_{\mathcal{D}}$ and $\mathcal{L}_{\mathcal{G}}$ are the discriminator and generator loss functions respectively, $\mathbb{P}$ is the distribution of target images I_{HR}, $\mathbb{Q}$ is the distribution of input images I_{LR}, $\mathcal{D}(.)$ and $\mathcal{G}(.)$ are discriminator and generator evaluated at I_{HR} and I_{LR} respectively. Having trained the discriminator $\mathcal{D}(.)$ to its optimal point, the loss function is an approximation of the Jensen-Shannon divergence (JSD) for Standard GAN [14]. Thus an interpretation of minimizing the generator loss $\mathcal{G}(.)$ can be roughly interpreted as minimizing the approximated divergence.

When Standard GAN is optimized, it is equivalent to the Jensen-Shannon divergence (JSD) [14]. JSD is minimized ($JSD(\mathbb{P}|\mathbb{Q}) = 0$), when $\mathcal{D}(I_{HR}) = \mathcal{D}(I_{SR}) = \frac{1}{2} \quad \forall \quad I_{HR} \in \mathbb{P}, I_{SR} \in \mathbb{Q}$ and maximized ($JSD(\mathbb{P}|\mathbb{Q}) = \log(2)$), when $\mathcal{D}(I_{HR}) = 1, \mathcal{D}(I_{SR}) = 0 \quad \forall \quad I_{HR} \in \mathbb{P}, I_{SR} \in \mathbb{Q}$. Hence, directly minimizing divergence should result in smooth decrease from 1 to $\frac{1}{2}$ in $\mathcal{D}(I_{HR})$ and smooth increase from 0 to $\frac{1}{2}$ in $\mathcal{D}(I_{SR})$. However, in Standard GAN, minimizing the loss only increases $\mathcal{D}(I_{SR})$ closer to 1 instead of $\frac{1}{2}$ and does not decrease $\mathcal{D}(I_{HR})$. This demonstrates the working of standard GAN is not the same as minimizing JSD. The loss functions of Relativistic average Discriminator (RaD) can be formulated as:

$$\mathcal{L}_{\mathcal{G}}^{Ra} = \mathbb{E}_{I_{HR} \sim \mathbb{P}} \left[\log \left(1 - \mathcal{D}_{Ra} \left(I_{HR} \right) \right) \right] + \mathbb{E}_{I_{LR} \sim \mathbb{Q}} \left[\log \left(\mathcal{D}_{Ra} \left(\mathcal{G} \left(I_{LR} \right) \right) \right) \right], \tag{3.40}$$

$$\mathcal{L}_{\mathcal{D}}^{Ra} = \mathbb{E}_{I_{HR} \sim \mathbb{P}} \left[\log \left(\mathcal{D}_{Ra} \left(I_{HR} \right) \right) \right] + \mathbb{E}_{I_{LR} \sim \mathbb{Q}} \left[\log \left(1 - \mathcal{D}_{Ra} \left(\mathcal{G} \left(I_{LR} \right) \right) \right) \right], \tag{3.41}$$

where $\mathcal{L}_{\mathcal{D}}^{Ra}$ and $\mathcal{L}_{\mathcal{G}}^{Ra}$ are discriminator and generator loss respectively,

$$\mathcal{D}_{Ra}(I) = \begin{cases} \sigma \left(C(I) - \mathbb{E}_{I_{SR} \sim \mathbb{Q}} \left[C \left(I_{SR} \right) \right) \right], & if \quad I = I_{HR} \\ \sigma \left(C(I) - \mathbb{E}_{I_{HR} \sim \mathbb{P}} \left[C \left(I_{HR} \right) \right) \right], & if \quad I = I_{SR} \end{cases} \tag{3.42}$$

where, σ is the sigmoid function, $C(.)$ is the non-transformed discriminator output as discussed above. In Standard GAN, the discriminator can be defined as $\mathcal{D}(I) = \sigma(C(I))$. A discriminator can be defined as relative by enforcing it to be dependant on both real and generated image pairs $\tilde{I} = (I_{HR}, I_{SR})$. The discriminator can thus be defined as $\mathcal{D}(\tilde{I}) = \sigma(C(I_{HR}) - C(I_{SR}))$. This modification of discriminator can be interpreted as: for the given generated image, the discriminator estimates the probability that the given real image is more realistic. Likewise, we can define $\overline{\mathcal{D}}(\tilde{I}) = \sigma(C(I_{SR}) - C(I_{HR}))$ as the probability of the generated image being more realistic than given real image. The discriminator $\overline{\mathcal{D}}$ is not required to be explicitly used in the loss function as $\log(1 - \overline{\mathcal{D}})(\tilde{I})$, due to the property:

$$\begin{aligned} 1 - \overline{\mathcal{D}}(\tilde{I}) &= 1 - \sigma(C(I_{SR}) - C(I_{HR})) \\ &= \sigma(C(I_{HR}) - C(I_{SR})) = \mathcal{D}(\tilde{I}) \\ &\therefore \log(\mathcal{D}(\tilde{I})) = \log(1 - \overline{\mathcal{D}}(\tilde{I})). \end{aligned} \tag{3.43}$$

The generator adversarial loss $\mathcal{L}_{a}^{SR}$ Eq. (3.35) comprises of $\mathcal{L}_{ga}^{SR_i}$ which is the generator adversarial loss in the image domain, defined as:

$$\mathcal{L}_{ga}^{SR_i} = -\mathbb{E}_{I_{HR}\sim\mathbb{P}}\left[\log(1 - \mathcal{D}_{Ra}^{i}\left(I_{HR}\right))\right] - \mathbb{E}_{I_{LR}\sim\mathbb{Q}}\left[\left(\log(\mathcal{D}_{Ra}^{i}\left(\mathcal{G}(I_{LR})\right))\right)\right], \tag{3.44}$$

and $\mathcal{L}_{ga}^{SR_{f_\ell}}$ which is the generator adversarial loss in the feature domain, defined as:

$$\mathcal{L}_{ga}^{SR_{f_\ell}} = -\mathbb{E}_{I_{HR}\sim\mathbb{P}}\left[\log(1 - \mathcal{D}_{Ra}^{f}\left(\Psi^{(\ell)}\left(I_{HR}\right)\right))\right] - \mathbb{E}_{I_{LR}\sim\mathbb{Q}}\left[\left(\log(\mathcal{D}_{Ra}^{f}\left(\Psi^{(\ell)}\left(\mathcal{G}(I_{LR})\right)\right))\right)\right], \tag{3.45}$$

where $\mathcal{D}_{Ra}^{i}$ and $\mathcal{D}_{Ra}^{f}$ are the Relativistic average Discriminators in image and feature domain respectively. The generator adversarial loss benefits by using both the high-resolution I_{HR} and super-resolved I_{SR} images, while SRGAN [13] and SRFeat [77] only use the the super-resolved image I_{SR}.

Likewise the discriminator networks $\mathcal{L}_{da}^{SR_i}$ and $\mathcal{L}_{da}^{SR_{f_\ell}}$ are the image and feature domain discriminators respectively, defined as:

$$\mathcal{L}_{da}^{SR_i} = -\mathbb{E}_{I_{HR}\sim\mathbb{P}}\left[\log(\mathcal{D}_{Ra}^{i}\left(I_{HR}\right))\right] - \mathbb{E}_{I_{LR}\sim\mathbb{Q}}\left[\left(\log(1 - \mathcal{D}_{Ra}^{i}\left(\mathcal{G}(I_{LR})\right))\right)\right], \tag{3.46}$$

$$\mathcal{L}_{da}^{SR_{f_\ell}} = -\mathbb{E}_{I_{HR}\sim\mathbb{P}}\left[\log(\mathcal{D}_{Ra}^{f}\left(\Psi^{(\ell)}\left(I_{HR}\right)\right))\right] - \mathbb{E}_{I_{LR}\sim\mathbb{Q}}\left[\left(\log(1-\mathcal{D}_{Ra}^{f}\left(\Psi^{(\ell)}\left(\mathcal{G}(I_{LR})\right)\right))\right)\right], \tag{3.47}$$

Relativistic average Discriminator has a more similar interpretation to that in Standard GAN. Both the discriminators estimate the probability of the given real data being more realistic than the fake data on average.

As the Standard GAN has been discussed in the Background of this work, the important part to note is, with respect to Relativistic Discriminator, the Standard GAN ignores the first term of the equation Eq (3.47) while training the generator, as the gradient is zero as the generator does not influence it.

$$\mathbb{E}_{I_{HR}\sim\mathbb{P}}\left[log\left(1-\mathcal{D}\left(I_{HR}\right)\right)\right] \tag{3.48}$$

On the contrary in Relativistic GAN the first term is influenced by the super-resolved image, since the $\mathcal{D}_{Ra}(.)$ function makes use of both, the real and generated image.

$$\mathbb{E}_{I_{HR}\sim\mathbb{P}}\left[\log\left(1-\mathcal{D}_{Ra}\left(I_{HR}\right)\right)\right] \tag{3.49}$$

$$\mathcal{D}_{Ra}(I_{HR}) = \sigma\left(C(I_{HR}) - \mathbb{E}_{I_{SR}\sim\mathbb{Q}}\left[C\left(\mathcal{G}\left(I_{LR}\right)\right)\right]\right) \tag{3.50}$$

In the proposed training method a pixel-wise loss and a VGG [72] based loss is used in the image domain. Correspondingly an image domain based feature domain based adversarial losses are also proposed using Relatives tic Discriminator. The advantage of using the feature domain, is the representation of abstract image structures as features which encourage the generator to produce more realistic structural high-frequency information instead of adding noise artifacts. To further prove the effectiveness of the proposed adversarial learning the following experiments were conducted.

The above proposed generator network baseline was used to demonstrate the experimental results. In particular four experiments were conducted to demonstrate the superiority of the proposed method. First the generator baseline was trained using a Standard GAN (SRGAN) [13] , second, a Relativistic GAN as seen in ESRGAN [6] was used, followed by a feature domain discriminator was used as seen in SRFeat [77] and finally the proposed Relativistic Discriminator in image and Feature domain. The experiments were run on 64 × 64 random crop of DIV2K [81] with validation on Set5 [37].

Network	PSNR	SSIM
SRGAN + baseline	33.5009	0.9236
ESRGAN + baseline	33.9271	0.9280
SRFeat + baseline	33.9662	0.9285
proposed learning + baseline	34.4696	0.9301

Table 3.3: Adversarial Learning Baseline Comparison

As it can be seen from [Table 3.3], the PSNR and SSIM values of the proposed baseline along with the proposed adversarial learning renders the best results. Further a qualitative comparison is shown in [Figure 3.9] on images from BSD100. Even with the qualitative comparison, it can be observed that the proposed network architecture performs the best.

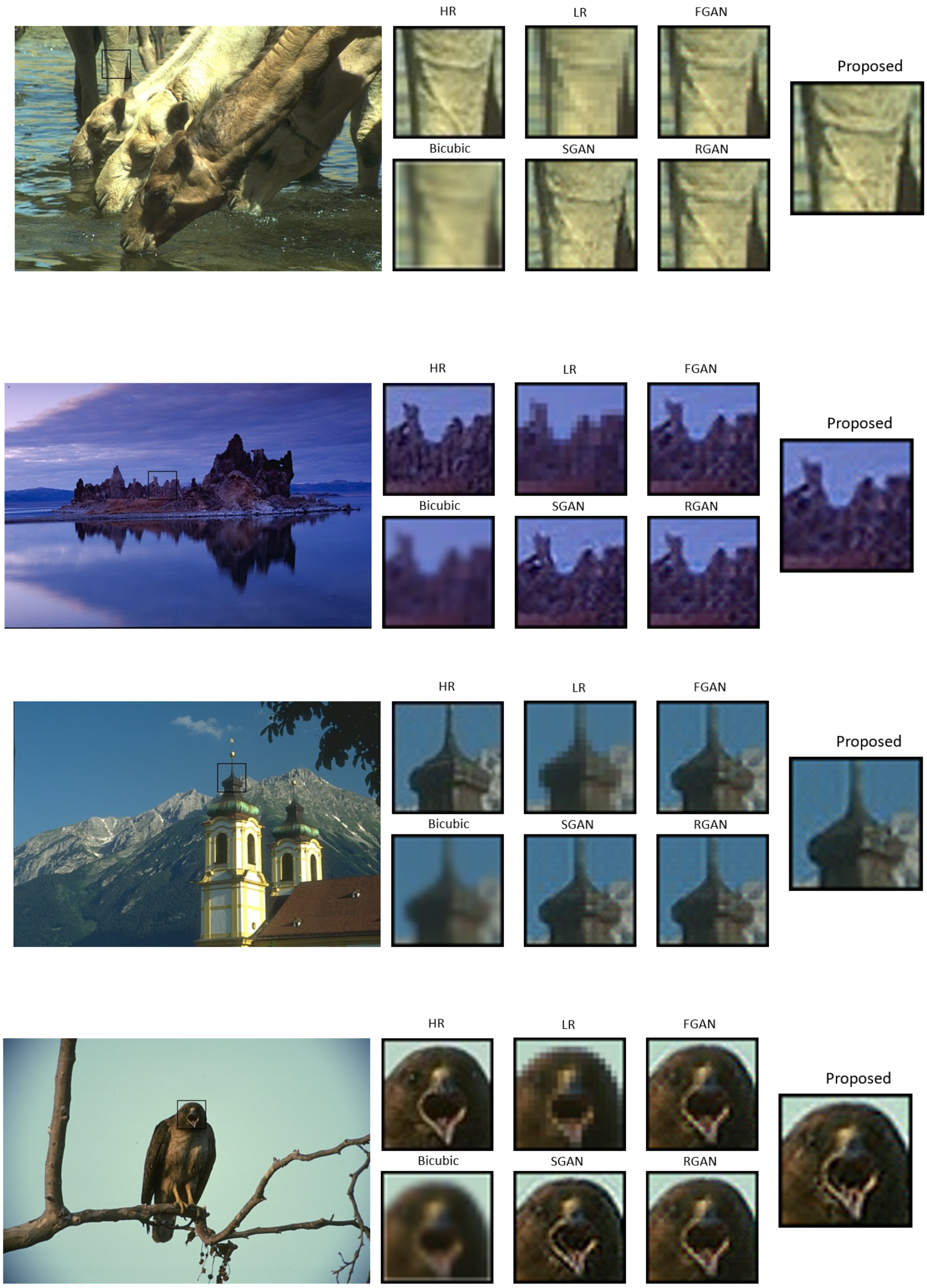

Figure 3.9: Qualitative comparison of proposed adversarial learning on 2x upscale

Chapter 4

Experimental Results

4.1 Overview

In the previous chapter the proposed network Architecture has been defined. This chapter would discuss the implementation details, training details, choice of dataset. Finally the results obtained by the proposed network are compared against some of the state-of-the-art networks and a detailed analysis is provided for the same.

4.2 Training Environment

The environment setup for all the conducted experiments are laid down in this section. All systems experimented on are based on Windows 10 operating system. In total five systems were used in parallel to run experiments with various state-of-the-art architectures. Furthermore, the requirement of multiple systems was critical due to the nature of this research, where each machine ran a variant of the same network architecture to find out the optimal components that contribute the most towards the best results. Each system was equipped with a GTX 1080 GPU with 8GB video memory, an i7 7700K @ 4.20GHZ with 16 GB memory.

4.3 Training Details

In the proposed network, there are two network settings that are suggested. The first setting is suggested to run the baseline as described in the previous chapter, which demonstrates real-time performance for upscaling 1280×720 images by 2× to achieve a 2K image (2560×1440) with 16 feature maps. The second setting demonstrated makes use of 10 Sub-Pixel Back-projection groups with 64 feature maps.

As discussed in the introduction, all experiments were performed with a 2× scaling factor between low-resolution and high-resolution. The low-resolution images are obtained by downsampling high-resolution images using scipy library in Python using bicubic linear interpolation. The training mini-batch size is set to 16 with low-resolution image crop of size 48×48 for the deep version of the network and mini-batch size of 32 with the same low-resolution image crop for the smaller network. An observation made in this process was, training the network with larger crop patch showed better results. To keep the comparisons with other networks fair, this crop size was selected. The training was done in two phases; first, the proposed generator network was trained to analyze its potential. Followed by training the network using the proposed feature and image-based Relativistic average Discriminator.

The proposed network was also trained using adversarial learning with low-resolution image crop of 64×64 with a mini-batch size of 8. A VGG-19 network was used for the discriminator network. The Generator network was trained using the L1 loss to maximize the PSNR. The model is trained using the ADAM optimizer for 1000 epochs, with $\beta_1 = 0.9$, $\beta_2 = 0.999$ and mini-batch repeating twice. The learning rate is initialized as 10^{-4} and decayed by a factor of 2 in every 200 epochs. No image augmentation has been used for training. When using adversarial learning, we set $\alpha = 10^{-2}$, which is the L1 pixel-wise image loss weight, $\gamma = 1$, which is the L1 pixel-wise feature loss weight. The adversarial weight λ is set to 10^{-3}.

4.4 Rendered Dataset

As discussed in the background section, there are few standard datasets used for training and validation in the research field of image super-resolution. Following the same for the convenience of comparison, all models have been trained using the DIV2K [81] dataset, which comprises of 800 training images. For validation Set5 [37], Set14 [80], BSD100 [78] and DIV2K [81] have been used. Since the focus of the research aims to demonstrate the applicability of real-time single image super-resolution in modern games, it can be argued that models trained and validated on natural images might not perform the same way on rendered or synthesized images. To observer how models perform on synthesized image vs. natural images, a new dataset has been introduced. This proposed dataset comprises of 800 2K images which contain rendered content from various games. The validation data set comprises of 200 2K images of game rendered content. This dataset was created by scraping the web for high-resolution game rendered content. Initially, a total of 2000+ images were downloaded. All content from the scraped images were carefully manually analyzed and selected to avoid duplicates, filtering for copyright watermarks and images with heads up displays (HUD), which do not reflect the actual content that would need to be super-resolved in a real-world scenario of super-resolving a game in real-time.

4.5 Evaluation

To validate the potential of the proposed network, several experiments and analysis have been performed. The proposed generator network is compared against seven state-of-the-art models: EDSR [30], VDSR [21], RCAN [4], DBPN [1], [10], FSRCNN [24] and ESPCN [25]. An extensive experiment has been conducted using 3 datasets for the generator network. All three datasets are of natural images. The evaluation parameters involves PSNR, SSIM and inference time in seconds. Higher PSNR and SSIM values indicate better quality. As a standard practice by all networks in comparison, all measurements use only the luminance channel (Y).

The proposed16 is the lightweight version of the generator network with 16 feature-maps with 1 group in Sub-Pixel Back-projection, likewise proposed64 uses 64 feature-maps with 10 groups in Sub-Pixel Back-project. This can be seen in [Table 4.1] for a numerical representation, and a qualitative comparison in [Figure 4.3] Furthermore, a Bubble chart [Figure 4.1] illustrates the inference time (in seconds) vs. SSIM vs. parameter count for these networks. The bubble chart evaluation is done on based on the BSDS100 dataset only.

The proposed adversarial network proposed in this research has been evaluated using the Set5 dataset. Due to lack of time and computational resources, the comparison was possible only against SRGAN [13] [Figure 4.2]. Further detailed comparison can be found in the supplementary material.

As discussed above, some experiments have also been performed on the new rendered image content dataset. The proposed lightweight network was trained using both natural and rendered images. An observation made was that, there was hardly any difference seen in the super-resolved image when trained with the rendered image or natural image. However, another interesting observation made was, even though the proposed adversarial network outperforms the state-of-the-art networks, training a generator with very few parameters isn't very helpful, since it adds more artifacts to the image rather than enhancing them. In [Figure 4.4], the images labeled as "Render" are generated using the lightweight network with 16 feature maps using the rendered image dataset. The images labeled as "Natural" are generated using the same lightweight network using DIV2K dataset. Images labeled as "GAN" are images trained on rendered dataset using the lightweight generator and the proposed adversarial network.

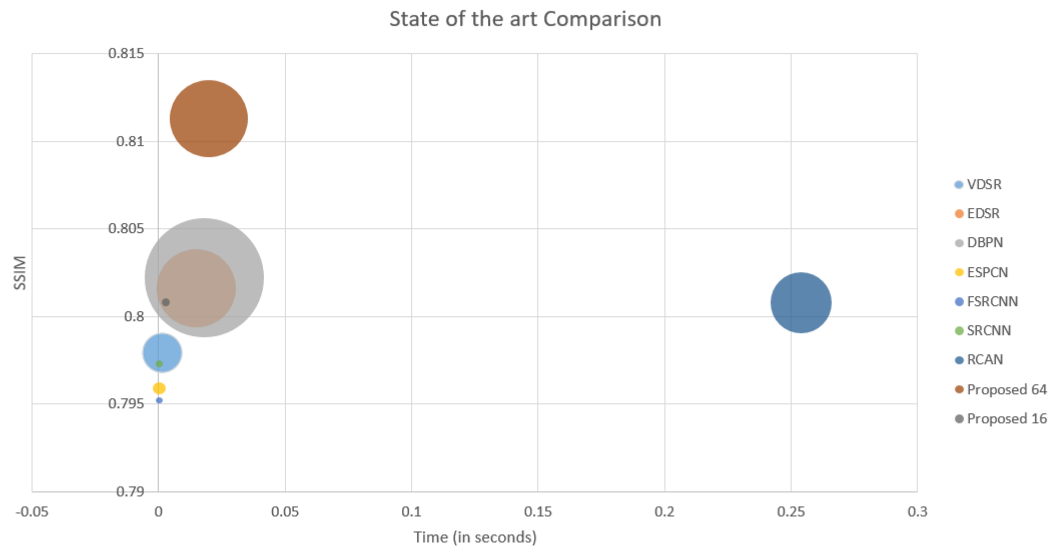

Figure 4.1: Bubble Graph comparing Quality VS. Inference Time vs. Parameter on 2x super-resolution

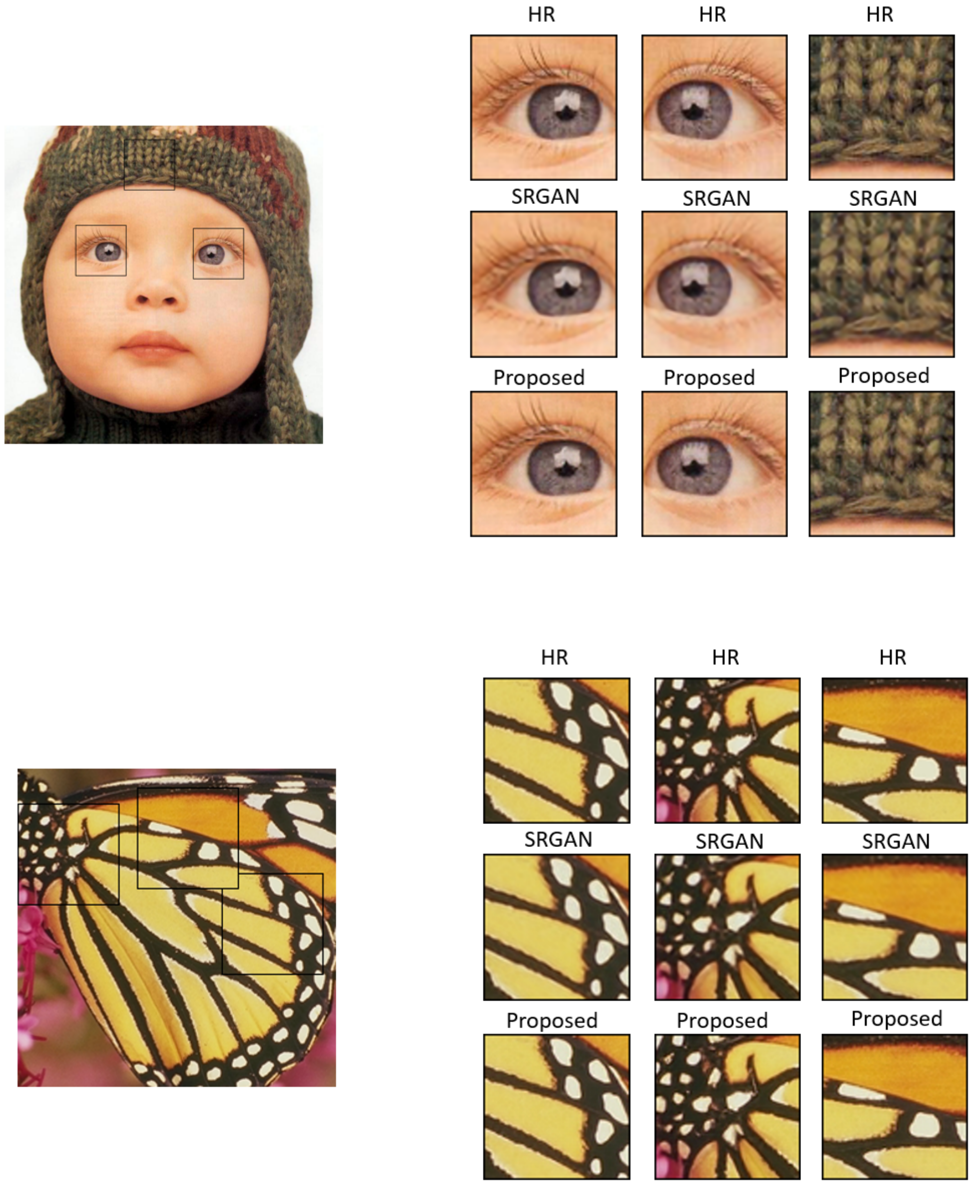

Figure 4.2: Qualitative comparison of proposed adversarial training vs. SRGAN on 2x super-resolution

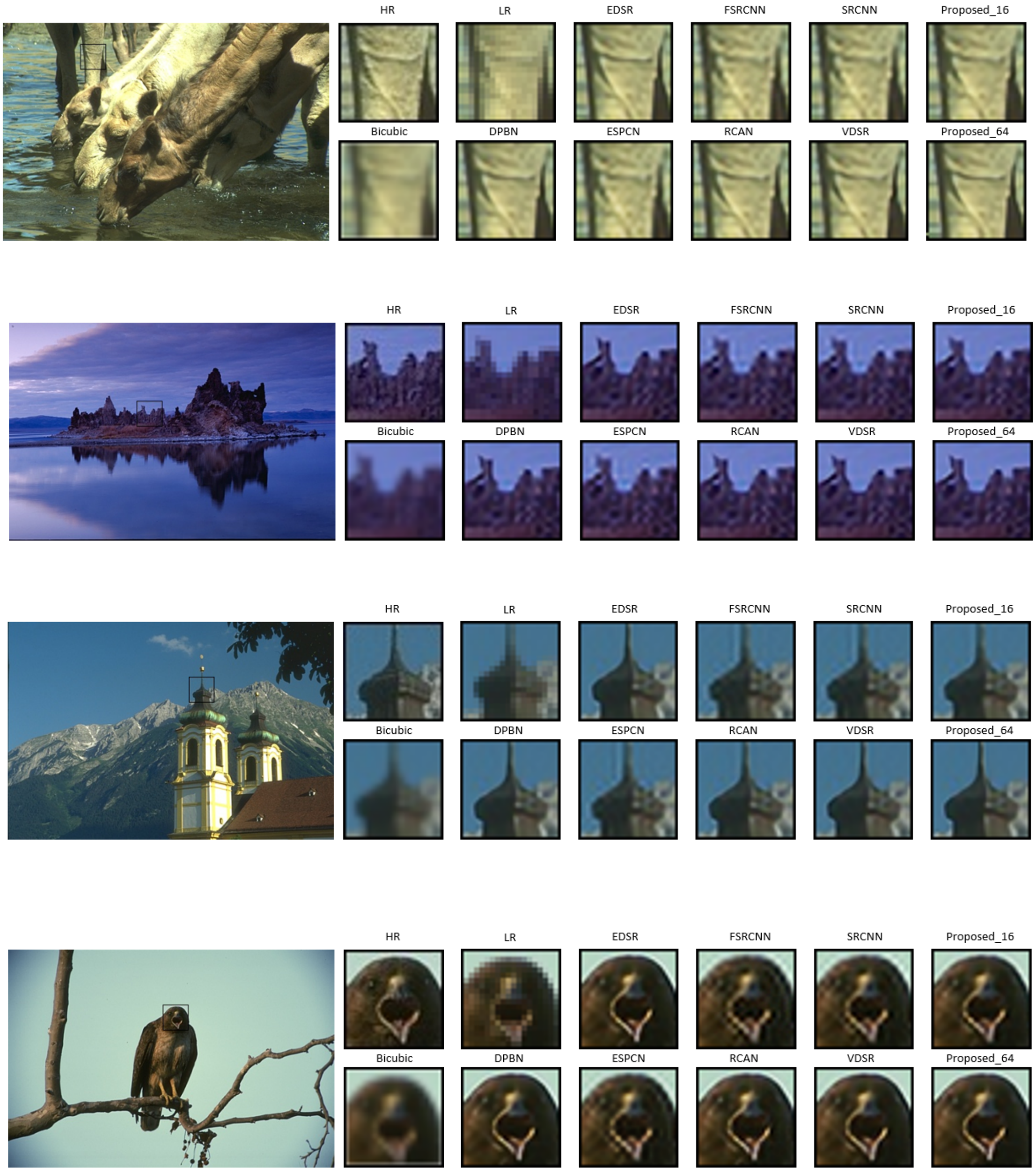

Figure 4.3: Qualitative comparison of proposed generator vs. SOTA on 2x super-resolution

Network	Dataset	PSNR	SSIM
DBPN	Set5	37.5171	0.9573
DBPN	Set14	31.8955	0.9018
DBPN	BSDS100	26.9452	0.8022
RCAN	Set5	37.5335	0.9564
RCAN	Set14	31.703	0.9027
RCAN	BSDS100	26.897	0.8008
EDSR	Set5	37.6979	0.9581
EDSR	Set14	31.92	0.9021
EDSR	BSDS100	26.9010	0.8016
VDSR	Set5	36.9455	0.9550
VDSR	Set14	31.5199	0.896
VDSR	BSDS100	26.8497	0.7976
SRCNN	Set5	35.2727	0.9422
SRCNN	Set14	30.8975	0.8882
SRCNN	BSDS100	27.0871	0.7973
FSRCNN	Set5	35.2744	0.9417
FSRCNN	Set14	30.7377	0.8858
FSRCNN	BSDS100	27.0923	0.7952
ESPCN	Set5	35.7220	0.9452
ESPCN	Set14	30.9505	0.8899
ESPCN	BSDS100	27.0318	0.7959
Proposed16	Set5	36.5512	0.9528
Proposed16	Set14	31.3425	0.8943
Proposed16	BSDS100	27.0567	0.8010
Proposed64	Set5	37.7546	0.9528
Proposed64	Set14	32.0108	0.903
Proposed64	BSDS100	27.2792	0.8113

Table 4.1: Comparison of proposed generator vs. SOTA on 2x super-resolution

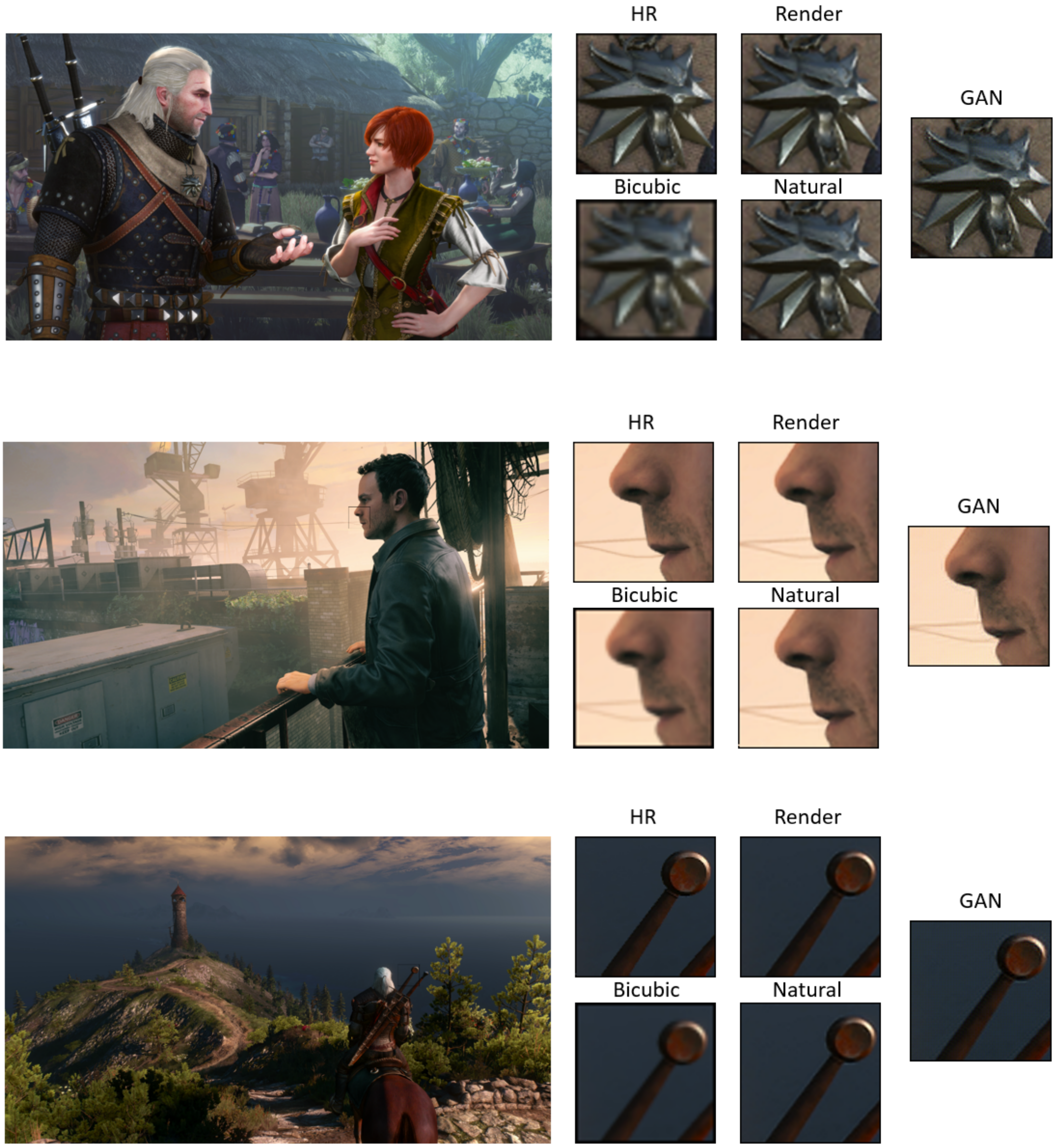

Figure 4.4: Qualitative comparison of Render vs Natural trained network on 2x super-resolution

Chapter 5

Conclusion and Future Work

5.1 Conclusion

This research demonstrates results on par with state-of-the-art. The primary research question, to find a suitable network architecture that is very shallow but doesn't sacrifice on performance has been presented in this work. The hypothesis formed at the beginning of this thesis has been proved with several experiments. Based on these experiments, few arguments have been formed, which have been validated by the final performance of the network. Despite proposing a novel adversarial network, it is worth to be noted that training a GAN is extremely difficult. Thus, even though the results are promising, it requires further work to find optimum values for the parameters in the loss function to train the adversarial network even better.

Since the proposed generator network has been demonstrated in two variants. The baseline being a very shallow network is suitable for applications like real-time rendering, where it can be used in modern game engines by integrating into the render pipeline directly. Since Microsoft has recently released DirectML, which allows integrating deep learning network directly into the DirectX 12 rendering pipeline, an implementation of this network into a rendering engine will be a promising start.

5.2 Future Work

Even though single image super-resolution is a research area that has been extensively explored for decades, there are still a lot of possibilities that can be explored to advance the field of real-time image super-resolution. Some of the possible future scope are listed below.

To improve real-time performance, shallow networks are a good solution. Another possible solution would be to train a full-sized network to achieve maximum quality possible. Once this network is trained, the weights can be pruned to improve the inference time.

Another interesting avenue where deep learning-based super-resolution has recently been introduced is Omnidirectional Images [95]. Since virtual reality is a popular area of research, real-time performance in virtual reality is something that still needs to be explored.

The experiments conducted with the adversarial network are still limited in nature. To be absolute about the networks perfomance intensive testing would be needed. In particular, testing with higher upscale factors would be a good point to start as only 2x results are tested.

Bibliography

[1] M. Haris, G. Shakhnarovich, and N. Ukita, "Deep back-projection networks for super-resolution," in *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, June 2018, pp. 1664–1673.

[2] W. Yang, X. Zhang, Y. Tian, W. Wang, J. Xue, and Q. Liao, "Deep learning for single image super-resolution: A brief review," *IEEE Transactions on Multimedia*, pp. 1–1, 2019.

[3] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 770–778.

[4] Y. Zhang, K. Li, K. Li, L. Wang, B. Zhong, and Y. Fu, "Image super-resolution using very deep residual channel attention networks," in *Computer Vision – ECCV 2018*, V. Ferrari, M. Hebert, C. Sminchisescu, and Y. Weiss, Eds. Cham: Springer International Publishing, 2018, pp. 294–310.

[5] Z. Li, J. Yang, Z. Liu, X. Yang, G. Jeon, and W. Wu, "Feedback network for image super-resolution," in *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2019.

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

[7] A. Edelsten. Nvidia dlss: Your questions, answered. [Online]. Available: https://www.nvidia.com/en-us/geforce/news/nvidia-dlss-your-questions-answered/

[8] E. Haines and T. Akenine-Möller, Eds., *Ray Tracing Gems.* Apress, 2019, http://raytracinggems.com.

[9] Nvidia. Dlss: What does it mean for game developers? [Online]. Available: https://news.developer.nvidia.com/dlss-what-does-it-mean-for-game-developers/

[10] C. Dong, C. C. Loy, K. He, and X. Tang, "Image super-resolution using deep convolutional networks," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 38, no. 2, pp. 295–307, Feb 2016.

[11] W. Shi, J. Caballero, L. Theis, F. Huszar, A. P. Aitken, C. Ledig, and Z. Wang, "Is the deconvolution layer the same as a convolutional layer?" *CoRR*, vol. abs/1609.07009, 2016. [Online]. Available: http://arxiv.org/abs/1609.07009

[12] A. Jolicoeur-Martineau, "The relativistic discriminator: a key element missing from standard GAN," in *International Conference on Learning Representations*, 2019. [Online]. Available: https://openreview.net/forum?id=S1erHoR5t7

[13] C. Ledig, L. Theis, F. Huszr, 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," in *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, July 2017, pp. 105–114.

[14] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, "Generative adversarial nets," in *Advances in Neural Information Processing Systems 27*, Z. Ghahramani, M. Welling, C. Cortes, N. D. Lawrence, and K. Q. Weinberger, Eds. Curran Associates, Inc., 2014, pp. 2672–2680. [Online]. Available: http://papers.nips.cc/paper/5423-generative-adversarial-nets.pdf

[15] M. Irani and S. Peleg, "Improving resolution by image registration," *CVGIP: Graphical Models and Image Processing*, vol. 53, pp. 231–239, 05 1991.

[16] Sung Cheol Park, Min Kyu Park, and Moon Gi Kang, "Super-resolution image reconstruction: a technical overview," *IEEE Signal Processing Magazine*, vol. 20, no. 3, pp. 21–36, May 2003.

[17] M. Protter, M. Elad, H. Takeda, and P. Milanfar, "Generalizing the nonlocal-means to super-resolution reconstruction," *IEEE Transactions on Image Processing*, vol. 18, no. 1, pp. 36–51, Jan 2009.

[18] K. Zhang, W. Zuo, and L. Zhang, "Learning a single convolutional super-resolution network for multiple degradations," in *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, June 2018, pp. 3262–3271.

[19] J. Schmidhuber, "Deep learning in neural networks: An overview," *Neural Networks*, vol. 61, pp. 85 – 117, 2015. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0893608014002135

[20] Y. Bengio, A. Courville, and P. Vincent, "Representation learning: A review and new perspectives," *IEEE transactions on pattern analysis and machine intelligence*, vol. 35, pp. 1798–1828, 08 2013.

[21] J. Kim, J. K. Lee, and K. M. Lee, "Accurate image super-resolution using very deep convolutional networks," in *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2016.

[22] ——, "Deeply-recursive convolutional network for image super-resolution," in *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2016.

[23] Y. Tai, J. Yang, and X. Liu, "Image super-resolution via deep recursive residual network," in *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Honolulu, HI, July 2017.

[24] C. Dong, C. C. Loy, and X. Tang, "Accelerating the super-resolution convolutional neural network," in *Computer Vision – ECCV 2016*, B. Leibe, J. Matas, N. Sebe, and M. Welling, Eds. Cham: Springer International Publishing, 2016, pp. 391–407.

[25] W. Shi, J. Caballero, F. Huszr, J. Totz, A. P. Aitken, R. Bishop, D. Rueckert, and Z. Wang, "Real-time single image and video super-resolution using an efficient sub-pixel convolutional neural network," in *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2016, pp. 1874–1883.

[26] T. Tong, G. Li, X. Liu, and Q. Gao, "Image super-resolution using dense skip connections," in *2017 IEEE International Conference on Computer Vision (ICCV)*, Oct 2017, pp. 4809–4817.

[27] W. Han, S. Chang, D. Liu, M. Yu, M. Witbrock, and T. S. Huang, "Image super-resolution via dual-state recurrent networks," in *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, June 2018, pp. 1654–1663.

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

[29] T. Dai, J. Cai, Y. Zhang, S.-T. Xia, and L. Zhang, "Second-order attention network for single image super-resolution," in *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2019.

[30] B. Lim, S. Son, H. Kim, S. Nah, and K. M. Lee, "Enhanced deep residual networks for single image super-resolution," in *2017 IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, July 2017, pp. 1132–1140.

[31] W. Lai, J. Huang, N. Ahuja, and M. Yang, "Deep laplacian pyramid networks for fast and accurate super-resolution," in *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, July 2017, pp. 5835–5843.

[32] ——, "Fast and accurate image super-resolution with deep laplacian pyramid networks," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, pp. 1–1, 2018.

[33] W. Yifan, F. Perazzi, B. McWilliams, A. Sorkine-Hornung, O. Sorkine-Hornung, and C. Schroers, "A fully progressive approach to single-image super-resolution," in *CVPR Workshops*, June 2018.

[34] Y. Bengio, J. Louradour, R. Collobert, and J. Weston, "Curriculum learning," in *Proceedings of the 26th Annual International Conference on Machine Learning*, ser. ICML '09. New York, NY, USA: ACM, 2009, pp. 41–48. [Online]. Available: http://doi.acm.org/10.1145/1553374.1553380

[35] G. Huang, Z. Liu, L. v. d. Maaten, and K. Q. Weinberger, “Densely connected convolutional networks,” in *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, July 2017, pp. 2261–2269.

[36] A. R. Zamir, T.-L. Wu, L. Sun, W. B. Shen, B. E. Shi, J. Malik, and S. Savarese, “Feedback networks,” in *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, July 2017.

[37] J. Yang, J. Wright, T. S. Huang, and Y. Ma, “Image super-resolution via sparse representation,” *Trans. Img. Proc.*, vol. 19, no. 11, pp. 2861–2873, Nov. 2010. [Online]. Available: http://dx.doi.org/10.1109/TIP.2010.2050625

[38] R. Timofte, V. DeSmet, and L. VanGool, “A+: Adjusted anchored neighborhood regression for fast super-resolution,” in *Computer Vision – ACCV 2014*, D. Cremers, I. Reid, H. Saito, and M.-H. Yang, Eds. Cham: Springer International Publishing, 2015, pp. 111–126.

[39] C. Yang and M. Yang, “Fast direct super-resolution by simple functions,” in *2013 IEEE International Conference on Computer Vision*, Dec 2013, pp. 561–568.

[40] S. Schulter, C. Leistner, and H. Bischof, “Fast and accurate image upscaling with super-resolution forests,” in *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2015, pp. 3791–3799.

[41] M. D. Zeiler, D. Krishnan, G. W. Taylor, and R. Fergus, “Deconvolutional networks,” in *In CVPRW*, 2010.

[42] X.-J. Mao, C. Shen, and Y.-B. Yang, “Image restoration using very deep convolutional encoder-decoder networks with symmetric skip connections,” in *Proceedings of the 30th International Conference on Neural Information Processing Systems*, ser. NIPS’16. USA: Curran Associates Inc., 2016, pp. 2810–2818. [Online]. Available: http://dl.acm.org/citation.cfm?id=3157382.3157412

[43] A. Odena, V. Dumoulin, and C. Olah, “Deconvolution and checkerboard artifacts,” *Distill*, 2016. [Online]. Available: http://distill.pub/2016/deconv-checkerboard/

[44] N. Ahn, B. Kang, and K.-A. Sohn, “Fast, accurate, and lightweight super-resolution with cascading residual network,” in *Computer Vision – ECCV 2018*, V. Ferrari, M. Hebert, C. Sminchisescu, and Y. Weiss, Eds. Cham: Springer International Publishing, 2018, pp. 256–272.

[45] M. Bevilacqua, A. Roumy, C. Guillemot, and M.-L. Alberi Morel, “Low-Complexity Single-Image Super-Resolution based on Nonnegative Neighbor Embedding,” in *British Machine Vision Conference (BMVC)*, Guildford, Surrey, United Kingdom, Sep. 2012. [Online]. Available: https://hal.inria.fr/hal-00747054

[46] R. Timofte, V. De, and L. V. Gool, “Anchored neighborhood regression for fast example-based super-resolution,” in *2013 IEEE International Conference on Computer Vision*, Dec 2013, pp. 1920–1927.

[47] K. He and J. Sun, “Convolutional neural networks at constrained time cost,” in *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2015, pp. 5353–5360.

[48] Z. Hui, X. Wang, and X. Gao, “Fast and accurate single image super-resolution via information distillation network,” in *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, June 2018, pp. 723–731.

[49] X. Glorot and Y. Bengio, “Understanding the difficulty of training deep feedforward neural networks,” in *Proceedings of the Thirteenth International Conference on Artificial Intelligence and Statistics*, ser. Proceedings of Machine Learning Research, Y. W. Teh and M. Titterington, Eds., vol. 9. Chia Laguna Resort, Sardinia, Italy: PMLR, 13–15 May 2010, pp. 249–256. [Online]. Available: http://proceedings.mlr.press/v9/glorot10a.html

[50] X. Li, Y. Sun, Y. Yang, and C. Miao, “Symmetrical residual connections for single image super-resolution,” *ACM Trans. Multimedia Comput. Commun. Appl.*, vol. 15, no. 1, pp. 19:1–19:10, Feb. 2019. [Online]. Available: http://doi.acm.org/10.1145/3282445

[51] Y. Tai, J. Yang, X. Liu, and C. Xu, “Memnet: A persistent memory network for image restoration,” in *Proceedings of International Conference on Computer Vision*, 2017.

[52] Z. Xingrong, L. Zheng, Z. Heming, and C. Xueqin, “Segmentation of right ventricular mr image based on deep neural network: Dilated densenet of two level losses,” in *2018 14th IEEE International Conference on Signal Processing (ICSP)*, Aug 2018, pp. 355–358.

[53] Y. Zhu and S. Newsam, “Densenet for dense flow,” in *2017 IEEE International Conference on Image Processing (ICIP)*, Sep. 2017, pp. 790–794.

[54] G. Ye, J. Ruan, C. Wu, J. Zhou, S. He, J. Wang, Z. Zhu, J. Yue, and Y. Zhang, “Multitask classification of breast cancer pathological images using se-densenet,” in *2019 Eleventh International Conference on Advanced Computational Intelligence (ICACI)*, June 2019, pp. 173–178.

[55] I. Elgendi, K. S. Munasinghe, and A. Jamalipour, “A three-tier sdn based distributed mobility management architecture for densenets,” in *2016 IEEE International Conference on Communications (ICC)*, May 2016, pp. 1–6.

[56] S. K. Choudhury, R. P. Padhy, and P. K. Sa, “Faster r-cnn with densenet for scale aware pedestrian detection vis--vis hard negative suppression,” in *2017 IEEE 27th International Workshop on Machine Learning for Signal Processing (MLSP)*, Sep. 2017, pp. 1–6.

[57] L. Wang, L. Qiu, W. Sui, and C. Pan, “Reconstructed densenets for image super-resolution,” in *2018 25th IEEE International Conference on Image Processing (ICIP)*, Oct 2018, pp. 3558–3562.

[58] B. Olshausen, C. Anderson, and D. Van Essen, “A neurobiological model of visual attention and invariant pattern recognition based on dynamic routing of information,” *Journal of Neuroscience*, vol. 13, no. 11, pp. 4700–4719, 1993. [Online]. Available: https://www.jneurosci.org/content/13/11/4700

[59] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. u. Kaiser, and I. Polosukhin, “Attention is all you need,” in *Advances in Neural Information Processing Systems 30*, I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, Eds. Curran Associates, Inc., 2017, pp. 5998–6008. [Online]. Available: http://papers.nips.cc/paper/7181-attention-is-all-you-need.pdf

[60] C. Cao, X. Liu, Y. Yang, Y. Yu, J. Wang, Z. Wang, Y. Huang, L. Wang, C. Huang, W. Xu, D. Ramanan, and T. S. Huang, "Look and think twice: Capturing top-down visual attention with feedback convolutional neural networks," in *2015 IEEE International Conference on Computer Vision (ICCV)*, Dec 2015, pp. 2956–2964.

[61] M. Jaderberg, K. Simonyan, A. Zisserman, and k. kavukcuoglu, "Spatial transformer networks," in *Advances in Neural Information Processing Systems 28*, C. Cortes, N. D. Lawrence, D. D. Lee, M. Sugiyama, and R. Garnett, Eds. Curran Associates, Inc., 2015, pp. 2017–2025. [Online]. Available: http://papers.nips.cc/paper/5854-spatial-transformer-networks.pdf

[62] L. Chen, H. Zhang, J. Xiao, L. Nie, J. Shao, W. Liu, and T.-S. Chua, "Sca-cnn: Spatial and channel-wise attention in convolutional networks for image captioning," in *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, July 2017.

[63] J. Hu, L. Shen, and G. Sun, "Squeeze-and-excitation networks," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2018.

[64] V. Nair and G. E. Hinton, "Rectified linear units improve restricted boltzmann machines," in *Proceedings of the 27th International Conference on International Conference on Machine Learning*, ser. ICML'10. USA: Omnipress, 2010, pp. 807–814. [Online]. Available: http://dl.acm.org/citation.cfm?id=3104322.3104425

[65] A. Dosovitskiy and T. Brox, "Generating images with perceptual similarity metrics based on deep networks," in *Advances in Neural Information Processing Systems 29*, D. D. Lee, M. Sugiyama, U. V. Luxburg, I. Guyon, and R. Garnett, Eds. Curran Associates, Inc., 2016, pp. 658–666. [Online]. Available: http://papers.nips.cc/paper/6158-generating-images-with-perceptual-similarity-metrics-based-on-deep-networks.pdf

[66] J. Johnson, A. Alahi, and L. Fei-Fei, "Perceptual losses for real-time style transfer and super-resolution," in *European Conference on Computer Vision*, 2016.

[67] H. Zhao, O. Gallo, I. Frosio, and J. Kautz, "Loss functions for image restoration with neural networks," *IEEE Transactions on Computational Imaging*, vol. 3, no. 1, pp. 47–57, March 2017.

[68] A. Bruhn, J. Weickert, and C. Schnörr, "Lucas/kanade meets horn/schunck: Combining local and global optic flow methods," *International Journal of Computer Vision*, vol. 61, no. 3, pp. 211–231, Feb 2005. [Online]. Available: https://doi.org/10.1023/B:VISI.0000045324.43199.43

[69] P. Charbonnier, L. Blanc-Feraud, G. Aubert, and M. Barlaud, "Two deterministic half-quadratic regularization algorithms for computed imaging," in *Proceedings of 1st International Conference on Image Processing*, vol. 2, Nov 1994, pp. 168–172 vol.2.

[70] M. S. M. Sajjadi, B. Schlkopf, and M. Hirsch, "Enhancenet: Single image super-resolution through automated texture synthesis," in *2017 IEEE International Conference on Computer Vision (ICCV)*, Oct 2017, pp. 4501–4510.

[71] X. Wang, K. Yu, C. Dong, and C. Change Loy, "Recovering realistic texture in image super-resolution by deep spatial feature transform," in *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2018.

[72] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," in *International Conference on Learning Representations*, 2015.

[73] Y. Yuan, S. Liu, J. Zhang, Y. Zhang, C. Dong, and L. Lin, "Unsupervised image super-resolution using cycle-in-cycle generative adversarial networks," in *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, June 2018, pp. 814–81 409.

[74] X. Mao, Q. Li, H. Xie, R. Y. K. Lau, Z. Wang, and S. P. Smolley, "Least squares generative adversarial networks," in *2017 IEEE International Conference on Computer Vision (ICCV)*, Oct 2017, pp. 2813–2821.

[75] A. Bulat, J. Yang, and G. Tzimiropoulos, "To learn image super-resolution, use a gan to learn how to do image degradation first," in *Computer Vision – ECCV*

2018, V. Ferrari, M. Hebert, C. Sminchisescu, and Y. Weiss, Eds. Cham: Springer International Publishing, 2018, pp. 187–202.

[76] T. Miyato, T. Kataoka, M. Koyama, and Y. Yoshida, "Spectral normalization for generative adversarial networks," in *International Conference on Learning Representations*, 2018. [Online]. Available: https://openreview.net/forum?id=B1QRgziT-

[77] S.-J. Park, H. Son, S. Cho, K.-S. Hong, and S. Lee, "Srfeat: Single image super-resolution with feature discrimination," in *The European Conference on Computer Vision (ECCV)*, September 2018.

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

[79] P. Arbelaez, M. Maire, C. Fowlkes, and J. Malik, "Contour detection and hierarchical image segmentation," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 33, no. 5, pp. 898–916, May 2011.

[80] R. Zeyde, M. Elad, and M. Protter, "On single image scale-up using sparse-representations," in *Curves and Surfaces*, J.-D. Boissonnat, P. Chenin, A. Cohen, C. Gout, T. Lyche, M.-L. Mazure, and L. Schumaker, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2012, pp. 711–730.

[81] E. Agustsson and R. Timofte, "Ntire 2017 challenge on single image super-resolution: Dataset and study," in *2017 IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, July 2017, pp. 1122–1131.

[82] J. Huang, A. Singh, and N. Ahuja, "Single image super-resolution from transformed self-exemplars," in *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2015, pp. 5197–5206.

[83] A. Fujimoto, T. Ogawa, K. Yamamoto, Y. Matsui, T. Yamasaki, and K. Aizawa, "Manga109 dataset and creation of metadata," in *Proceedings of the 1st International Workshop on coMics ANalysis, Processing and Understanding*, ser.

MANPU '16. New York, NY, USA: ACM, 2016, pp. 2:1–2:5. [Online]. Available: http://doi.acm.org/10.1145/3011549.3011551

[84] R. Zhang, P. Isola, A. A. Efros, E. Shechtman, and O. Wang, "The unreasonable effectiveness of deep features as a perceptual metric," in *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2018.

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

[86] Z. Wang, A. C. Bovik, and L. Lu, "Why is image quality assessment so difficult?" in *2002 IEEE International Conference on Acoustics, Speech, and Signal Processing*, vol. 4, May 2002, pp. IV–3313–IV–3316.

[87] H. R. Sheikh, M. F. Sabir, and A. C. Bovik, "A statistical evaluation of recent full reference image quality assessment algorithms," *IEEE Transactions on Image Processing*, vol. 15, no. 11, pp. 3440–3451, Nov 2006.

[88] Z. Wang and A. C. Bovik, "Mean squared error: Love it or leave it? a new look at signal fidelity measures," *IEEE Signal Processing Magazine*, vol. 26, no. 1, pp. 98–117, Jan 2009.

[89] A. Hore and D. Ziou, "Image quality metrics: Psnr vs. ssim," in *2010 20th International Conference on Pattern Recognition*, Aug 2010, pp. 2366–2369.

[90] Z. Wang, D. Liu, J. Yang, W. Han, and T. Huang, "Deep networks for image super-resolution with sparse prior," in *Proceedings of the 2015 IEEE International Conference on Computer Vision (ICCV)*, ser. ICCV '15. Washington, DC, USA: IEEE Computer Society, 2015, pp. 370–378. [Online]. Available: http://dx.doi.org/10.1109/ICCV.2015.50

[91] Jianchao Yang, J. Wright, T. Huang, and Yi Ma, "Image super-resolution as sparse representation of raw image patches," in *2008 IEEE Conference on Computer Vision and Pattern Recognition*, June 2008, pp. 1–8.

[92] G. Montúfar, R. Pascanu, K. Cho, and Y. Bengio, “On the number of linear regions of deep neural networks,” in *Proceedings of the 27th International Conference on Neural Information Processing Systems - Volume 2*, ser. NIPS’14. Cambridge, MA, USA: MIT Press, 2014, pp. 2924–2932. [Online]. Available: http://dl.acm.org/citation.cfm?id=2969033.2969153

[93] C. Szegedy, S. Ioffe, V. Vanhoucke, and A. A. Alemi, “Inception-v4, inception-resnet and the impact of residual connections on learning,” in *Proceedings of the Thirty-First AAAI Conference on Artificial Intelligence*, ser. AAAI’17. AAAI Press, 2017, pp. 4278–4284. [Online]. Available: http://dl.acm.org/citation.cfm?id=3298023.3298188

[94] Y. Chen, J. Li, H. Xiao, X. Jin, S. Yan, and J. Feng, “Dual path networks,” in *Advances in Neural Information Processing Systems 30*, I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, Eds. Curran Associates, Inc., 2017, pp. 4467–4475. [Online]. Available: http://papers.nips.cc/paper/7033-dual-path-networks.pdf

[95] C. Ozcinar, A. Rana, and A. Smolic, “Super-resolution of omnidirectional images using adversarial learning,” 09 2019.