UNIVERSIDADE FEDERAL DE PERNAMBUCO
CENTRO DE INFORMÁTICA
PROGRAMA DE PÓS-GRADUAÇÃO EM CIÊNCIA DA COMPUTAÇÃO

Kevin Ian Ruiz Vargas

UR-SRGAN: A Generative Adversarial Network for Real-world Super-resolution with a U-Net-based Discriminator

Recife
2022

Kevin Ian Ruiz Vargas

UR-SRGAN: A Generative Adversarial Network for Real-world Super-resolution with a U-Net-based Discriminator

Dissertação de Mestrado apresentada ao Programa de Pós-graduação em Ciência da Computação do Centro de Informática da Universidade Federal de Pernambuco, como requisito parcial para obtenção do grau de Mestre em Ciência da Computação.

Área de Concentração: Inteligência Computacional
Orientador: Prof. Dr. Tsang Ing Ren

Recife
2022

Catalogação na fonte
Bibliotecária Monick Raquel Silvestre da S. Portes, CRB4-1217

V297u Vargas, Kevin Ian Ruiz

UR-SRGAN: a generative adversarial network for real-world super-resolution with a U-Net-based discriminator / Kevin Ian Ruiz Vargas. – 2022.

89 f.: il., fig., tab.

Orientador: Tsang Ing Ren.

Dissertação (Mestrado) – Universidade Federal de Pernambuco. CIn, Ciência da Computação, Recife, 2022

Inclui referências e apêndice.

1. Inteligência computacional. 2. Resolução de imagem. 3. Modelagem. I. Ren, Tsang Ing (orientador). II. Título.

006.31 CDD (23. ed.) UFPE - CCEN 2022-166

Kevin Ian Ruiz Vargas

“UR-SRGAN: A Generative Adversarial Network for Real-world Super-resolution with a U-Net-based Discriminator”

Dissertação de Mestrado apresentada ao Programa de Pós-Graduação em Ciência da Computação da Universidade Federal de Pernambuco, como requisito parcial para a obtenção do título de Mestre em Ciência da Computação. Área de Concentração: Inteligência Computacional.

Aprovado em: 08/08/2022.

BANCA EXAMINADORA

____

Prof. Dr. George Darmiton da Cunha Cavalcanti
Centro de Informática / UFPE

____

Prof. Dr. Ing Jyh Tsang
IDLab / University of Antwerp - Bélgica

____

Prof. Dr. Tsang Ing Ren
Centro de Informática / UFPE
(**Orientador**)

Dedico este trabalho à minha mãe, Olivia Vargas. Graças a ela estou onde estou agora e sou a pessoa que sou.

ACKNOWLEDGEMENTS

This work was possible thanks to my mother Olivia Vargas, thanks to all her support, love, and sacrifice I was able to finish my studies in systems engineering to be part of this prestigious master's program. I thank my stepfather, Rolando Morales, and my whole family in general for always being present and supporting me in all my goals to be fulfilled. I also want to thank the professor Dr. Tsang Ing Ren, for all his guidance and teachings that he gave me from the beginning and the opportunity to work on innovative projects in the Motorola Mobility LLC group with the Halide team where I learned a lot to develop myself professionally. Also, I want to thank my girlfriend Fernanda who accompanied me and helped me throughout this stage and I thank all the good friends I had the opportunity to meet in Brazil and their sincere friendship. Last, but not least, I would like to thank all the professors and staff of the Center for Informatics (UFPE) for giving me the opportunity to be part of this prestigious program. I thank the postgraduate office staff who always helped me in all aspects of registration and arrival to Brazil. This work was possible with financial support from the CAPES Foundation and the research cooperation project between Motorola Mobility LLC (a Lenovo Company) and the Center for Informatics (UFPE).

ABSTRACT

Despite several improvements in Super-Resolution deep learning techniques, these proposed methods tend to fail in many real-world scenarios since their models are usually trained using a pre-defined degradation process from high-resolution (HR) ground truth images to low-resolution (LR) ones. In this work, we propose a supervised Generative Adversarial Network (GAN) model for Image Super-Resolution which has as the first stage to estimate blur kernels and noise estimation from real-world images to generate LR images for the training phase. Furthermore, the proposal includes implementing a novel U-Net-based discriminator, to consider an input image's global and local context, and it allows employing a CutMix data augmentation for consistency regularization in the two-dimensional output space of the decoder. The proposed model was applied to three main datasets that are ordinarily used in super-resolution official competitions. The commonly-used evaluation metrics for image restoration were used for this evaluation: Peak Signal-to-Noise Ratio (PSNR), Structural Similarity (SSIM), Learned Perceptual Image Patch Similarity (LPIPS) and Natural Image Quality Evaluator (NIQE). After implementing this new architecture, three other prominent models in the state-of-the-art GAN proposals for super-resolution were trained with the same parameters and databases to perform a global comparison between all of them. Finally, the results of the experimentation in training and evaluation tasks between all the models suggest an improvement in the performance of the presented work compared to the other architectures based on the established metrics.

Keywords: image super-resolution; deep learning; loss functions; degradation modelling.

RESUMO

Apesar de várias melhorias nas técnicas de aprendizado profundo de super-resolução, esses métodos propostos tendem a falhar em muitos cenários do mundo real, pois seus modelos geralmente são treinados usando um processo de degradação predefinido de imagens de verdade de alta resolução - *High Resolution* (HR) para baixa resolução - *Low Resolution* (LR). Neste trabalho, propomos um modelo supervisionado de *Generative Adversarial Network* (GAN) para Super-Resolução de Imagem que tem como primeira etapa estimar *kernels* de borramento e estimativa de ruído de imagens do mundo real para gerar imagens LR para a fase de treinamento. Além disso, a proposta inclui a implementação de um novo discriminador baseado em U-Net, para considerar o contexto global e local de uma imagem de entrada, e permite empregar um aumento de dados CutMix para regularização de consistência no espaço de saída bidimensional do decodificador. O modelo proposto foi aplicado a três conjuntos de dados principais que são normalmente usados em competições oficiais de super-resolução. As métricas de avaliação comumente usadas para restauração de imagem foram usadas para esta avaliação: *Peak Signal-to-Noise Ratio* (PSNR), *Structural Similarity* (SSIM), *Learned Perceptual Image Patch Similarity* (LPIPS) e *Natural Image Quality Evaluator* (NIQE). Após a implementação desta nova arquitetura, três outros modelos de destaque nas propostas GAN de super-resolução de última geração foram treinados com os mesmos parâmetros e bancos de dados para realizar uma comparação global entre todos eles. Por fim, os resultados da experimentação em tarefas de treinamento e avaliação entre todos os modelos sugerem uma melhora no desempenho do trabalho apresentado em relação às demais arquiteturas baseadas nas métricas estabelecidas.

Palavras-chaves: super-resolução de imagem; deep learning; funções de perda; modelagem de degradação.

LIST OF FIGURES

LIST OF TABLES

LIST OF ABBREVIATIONS AND ACRONYMS

BISR	Blind Image Super Resolution
BSR	Blind Super Resolution
CCD	Charge Coupled Device
CMOS	Complementary Metal Oxide Semiconductor
CNN	Convolutional Neural Network
DAN	Deep Alternate Network
DIV2K	DIVerse 2K resolution high quality images
DPED	DSLR Photo Enhancement Dataset
DPSR	Deep Plug-and-play Super Resolution
ESRGAN	Enhanced Super Resolution Generative Adversarial Network
GAN	Generative Adversarial Network
GT	Ground Truth
HR	High Resolution
IKC	Iterative Kernel Correction
IQA	Image Quality Assessment
ISR	Image Super Resolution
JPEG	Joint Photographic Experts Group
KMSR	Kernel Modeling Super Resolution
LPIPS	Learned Perceptual Image Patch Similarity
LR	Low Resolution
MA	Mult Adds
MISR	Multi Image Super Resolution
MSE	Mean Square Error
NIMA	Neural Image Assessment
NIQE	Natural Image Quality Evaluator
OST	Outdoor Scenes
PSNR	Peak Signal-to-Noise Ratio
RRDB	Residual in Residual Dense Block
RWSR	Real World Super Resolution

SISR	Single Image Super Resolution
SOTA	State-of-the-art
SR	Super Resolution
SRCNN	Super Resolution Convolutional Neural Network
SRGAN	Super Resolution Generative Adversarial Network
SRMD	Super Resolution for Multiple Degradations
SSIM	Structural Similarity
UDVD	Unified Dynamic Convolutional Network for Variational Degradations
VSR	Video Super Resolution
VVG	Visual Geometry Group

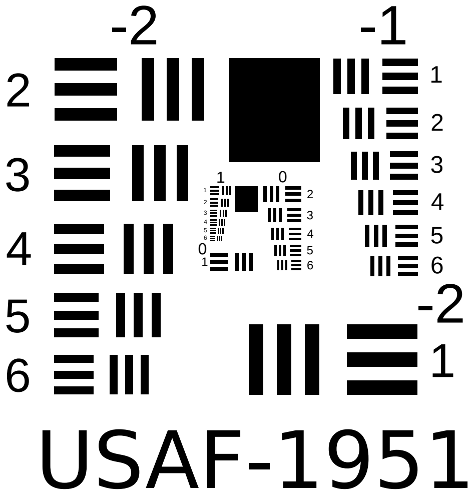
CONTENTS

1 INTRODUCTION

1.1 INTRODUCTION TO SINGLE-IMAGE SUPER-RESOLUTION

In most digital imaging applications, HR images or videos are usually desired for later image processing and analysis. The inclination for HR stems from two principal application areas: enhancement of pictorial information for human interpretation and helping representation for automatic machine perception (EL-SAMIE; HADHOUD; EL-KHAMY, 2019). Image resolution describes the details contained in an image. The higher the resolution, the more image details. The resolution of a digital image can be classified in several ways: pixel resolution, spatial resolution, spectral resolution, temporal resolution, and radiometric resolution. In the scope of this project, it will only be considered the spatial resolution type. Regarding digital images, spatial resolution concerns the number of small picture elements called pixels utilized in constructing the image. In other words, spatial resolution refers to the pixel density in an image and measures in pixels per unit area. Figure 1 exhibits a classic test target used in optical engineering laboratory work to analyze and validate imaging systems such as microscopes, cameras, and image scanners. Its purpose was to determine the spatial resolution of an imaging system.

Figure 1 – The 1951 USAF resolution test chart, is a classic test target used to identify the spatial resolution of imaging sensors and imaging systems.



Source: (MILANFAR, 2009)

Traditionally, conventional image vidicon and orthicon cameras have been the only available image acquisition devices. These cameras are analog. Since the 1970s, Charge Coupled Device (CCD) and Complementary Metal Oxide Semiconductor (CMOS) image sensors have been widely used to capture digital images. These sensors are typically arranged in a two-dimensional array to capture two-dimensional image signals. The sensor size or, equivalently, the number of sensor elements per unit area in the first place determines the spatial resolution of the image to capture (MILANFAR, 2009). The higher density of the sensors, the higher the spatial resolution possible of the imaging system. An imaging system with inadequate detectors will generate LR images with blocky effects due to the aliasing from low spatial sampling frequency.

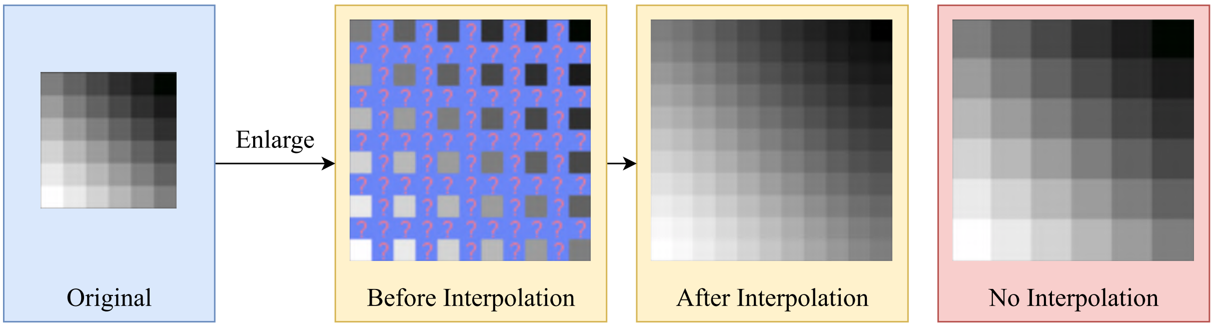
To increase the spatial resolution of an imaging system, one straightforward way is to increase the sensor density by reducing the sensor size. Nevertheless, as the sensor size decreases, the light incident on each sensor also decreases, causing the so-called shot noise. Also, the hardware cost of a sensor increases with the increase of sensor density or corresponding image pixel density. Therefore, the hardware limitation on the sensor size restricts the spatial resolution of an image that can be captured. While the image sensors limit the spatial resolution of the image, the image details (high-frequency bands) are also limited by the optics due to lens blurs, lens aberration effects, aperture diffractions, and optical blurring due to motion. Constructing imaging chips and optical components to capture very HR images is prohibitively expensive and impractical in most real applications, e.g., widely used surveillance cameras and cell phone built-in cameras. Besides the cost, the resolution of a surveillance camera is also limited by the camera speed and hardware storage.

The most feasible solution to this problem is integrating hardware and software capabilities to obtain the required HR level. Using as high an HR level as possible from the hardware can carry part of this task. The rest of the task is performed using the software. This is the new trend in the most up-to-date image-capturing devices. Image processing algorithms can effectively obtain HR images. Using a single LR image to obtain an HR image is known as image interpolation. On the other hand, when multiple degraded observations of the same scene are used to generate a single HR image, the process is known as Image Super Resolution (ISR), can be carried out using some a priori information about the degradations in the available LR images, such as information about blurring, registration shifts, and noise. With this information available, the solution to the SR reconstruction problem can be carried out easily. However, if no information is available, the problem is more complex, and it is known as Blind Image Super Resolution (BISR).

1.1.1 Image Interpolation

Image interpolation is the process by which a single HR image is obtained from a single LR image. Interpolation works by using known data to estimate values at unknown points and can be classified as polynomial interpolation and interpolation as an inverse problem. Figure 2 illustrates how resizing and enlargement work:

Figure 2 – Image interpolation for resizing or enlargement works in two directions and tries to achieve the best approximation of a pixel's color and intensity based on the values of surrounding pixels.



Source: The author (2022)

This method is the most widely used upsampling method. The current mainstream interpolation methods are Nearest-neighbor Interpolation, Bilinear Interpolation, and Bicubic Interpolation. Because these are highly interpretable and easy to implement, these methods are still widely used today.

- Nearest-neighbor Interpolation is a simple and intuitive algorithm that selects the nearest pixel value for each position to be interpolated, which has fast execution time but has difficulty producing high-quality results.
- Bilinear Interpolation sequentially performs linear interpolation operations on the two axes of the image. This method can obtain better results than nearest-neighbor interpolation while maintaining a relatively fast speed.
- Bicubic Interpolation performs cubic interpolation on each of the two axes. Compared with Bilinear, the results of Bicubic are smoother with fewer artifacts but slower than other interpolation methods.

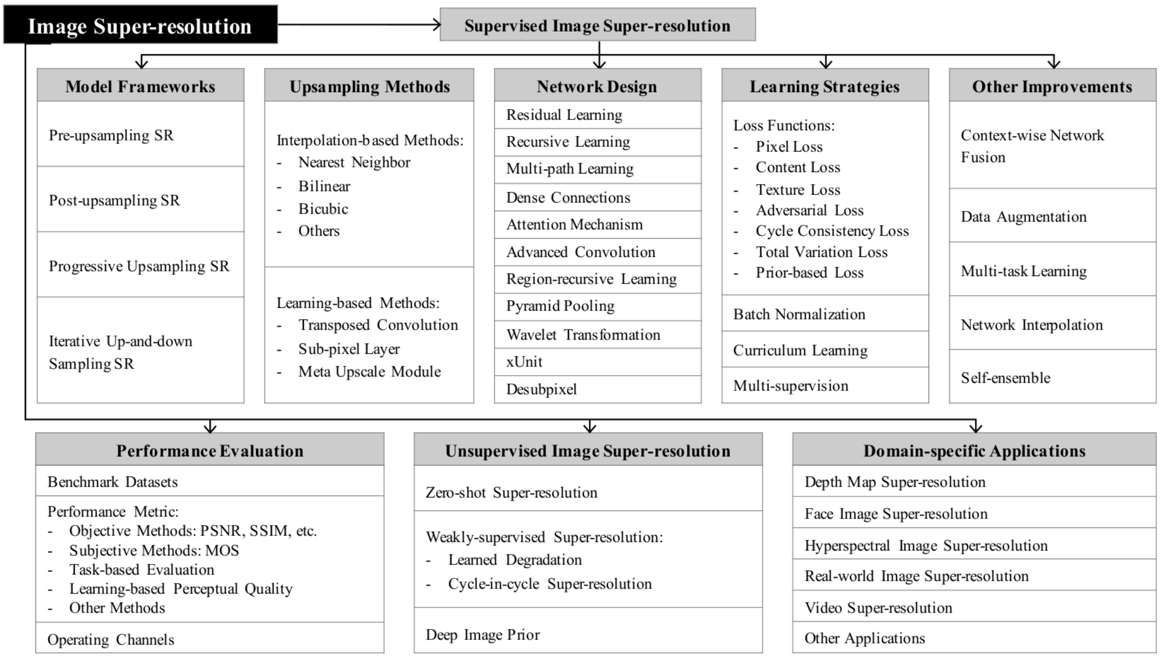
Interpolation is also the mainstream method for constructing (Single Image Super Resolution (SISR)) paired datasets and is widely used in the data pre-processing of Convolutional Neural Network (CNN)-based (SISR) models.

1.1.2 Image Super-resolution

ISR is the process of obtaining HR images from LR images. It is an important class of image processing techniques in computer vision and image processing. It enjoys a wide range of real-world applications, such as medical imaging, satellite imaging, surveillance and security, and astronomical imaging, amongst others. ISR techniques can be classified into SISR and Multi Image Super Resolution (MISR) according to the number of the input LR images. In particular, MISR has gradually developed into Video Super Resolution (VSR). Compared with MISR/VSR, SISR is much more challenging since MISR/VSR has extra information for reference while SISR only has information of a single input image for the missing image features reconstruction.

With the advancement in deep learning techniques in recent years, deep learning-based SR models have been actively explored and often achieve State-of-the-art (SOTA) performance on various benchmarks of SR. A variety of deep learning methods have been applied to solve SR tasks, ranging from the early CNN based method to recent promising Generative Adversarial Nets based SR approaches. (WANG; CHEN; HOI, 2020) provide a comprehensive survey on recent advances in ISR using deep learning approaches. Figure 3 shows the taxonomy of image SR that this survey covers in a hierarchically-structured way. This chapter will only include the most popular ones and, finally, provide more details about GAN methods that are the base architecture and more relevant for this thesis.

Figure 3 – Hierarchically-structured taxonomy of SR survey.



Source: (WANG; CHEN; HOI, 2020)

1.1.2.1 Pre-upsampling Super Resolution

This approach uses traditional techniques–like bicubic interpolation and deep learning–to refine an upsampled image. For example, Super Resolution Convolutional Neural Network (SRCNN) is a simple CNN architecture consisting of three layers: one for patch extraction, non-linear mapping, and reconstruction. The patch extraction layer is used to extract dense patches from the input and to represent them using convolutional filters. The non-linear mapping layer consists of 1 × 1 convolutional filters used to change the number of channels and add non-linearity. The final reconstruction layer returns the HR image

(DONG et al., 2015). Later on, Very Deep Super Resolution (VDSR) is an improvement on SRCNN, in which a deep network with small 3×3 convolutional filters is used instead of a smaller network with large convolutional filters (based on the Visual Geometry Group (VVG) architecture). This network tries to learn the residual of the output image and the interpolated input rather than learning the direct mapping (KIM; LEE; LEE, 2016a).

1.1.2.2 Post-upsampling Super-resolution

Since the feature extraction process in pre-upsampling SR occurs in the HR space, the computational power required is also on the higher end. Post-upsampling SR tries to solve this by doing feature extraction in the lower resolution space, then doing upsampling only at the end, therefore significantly reducing computation. Some techniques following this structure are the Fast Super-Resolution Convolutional Neural Networks (FSRCNN), a compact hourglass-shape CNN structure for fast image SR. With the collaboration of a set of deconvolution filters, the network can learn an end-to-end mapping between the original LR and HR images with no pre-processing. Their experiments show that it achieves a speed-up of more than 40× while still keeping its optimal performance (DONG; LOY; TANG, 2016) and Efficient Sub-Pixel Convolutional Neural Network (ESPCN), a CNN architecture with an efficient sub-pixel convolution layer which learns an array of upscaling filters to upscale the final LR feature maps into the HR output. As a result, the handcrafted bicubic filter is replaced in the SR pipeline with more complex upscaling filters trained explicitly for each feature map while also reducing the computational complexity of the overall SR operation (SHI et al., 2016).

1.1.2.3 Residual Networks

The EDSR architecture and its extension MDSR, with multiple input and output modules that give corresponding resolution outputs at $2\times, 3\times$, and $4\times$. A large kernel is used for the pre-processing layers to keep the network shallow while still achieving a high receptive field and pre-processing modules are the shared residual blocks, which is a standard block for data of all resolutions (LIM et al., 2017). Coming next, CARN is presented, a cascading mechanism at both the local and global level to incorporate features from multiple layers and give the network the ability to receive more information. These cascading modules effectively boost performance via multi-level representation and multiple shortcut connections. This work also proceeds to the CARN-M proposal for efficient SR by combining the efficient residual block and the recursive network scheme (AHN; KANG; SOHN, 2018).

1.1.2.4 Recursive Networks

Recursive networks employ shared network parameters in convolutional layers to reduce their memory footprints such as Deep Recursive Convolutional Networks (DRCN), this strategy repeatedly applies the same convolutional layer as often as desired. The number of parameters does not increase while more recursions are performed and improving the simple recursive network in two ways: recursive supervision and skip connection. In addition, it has a receptive field of 41 by 41, and this is relatively large compared to SRCNN (KIM; LEE; LEE, 2016c), and Deep Recursive Residual Network (DRRN), an improvement over DRCN by having residual blocks in the network over superficial convolutional layers. Specifically, residual learning is adopted, both in global and local manners, to mitigate the difficulty of training very deep networks; recursive learning is used to control the model parameters while increasing the depth. Their benchmark evaluation shows that DRRN significantly outperforms previous SOTA methods in SISR like VDSR, DRCN, and RED30, while utilizing far fewer parameters (TAI; YANG; LIU, 2017).

1.1.2.5 Attention-based Networks

The networks discussed so far give equal importance to all spatial locations and channels. In general, giving selective attention to different regions in an image can give much better results. Some architectures that help achieve this are SelNet, a 22-layered deep CNN structure, which can reconstruct HR images of higher quality with a slightly increased complexity, compared to the baseline only with ReLU. This solution was ranked in the 5th place in the NTIRE2017 Challenge, with much lower testing time compared to the top-4 entries. SelNet can separate hat strings, where other SR methods have difficulty (CHOI; KIM, 2017), and Residual Channel Attention Networks (RCAN), which feature extraction outcome is sent to the final layer with a long skip connection. Each residual group contains some blocks with short skip connections to carry the low-frequency signals from the LR image. At the same time, the leading network focuses on capturing high-frequency information. Furthermore, it includes a channel attention mechanism to adaptively rescale channel-wise features by considering interdependencies among channels (ZHANG et al., 2018).

1.1.2.6 Generative Models

The networks discussed so far optimize the pixel difference between predicted and output HR images. Even though this metric has satisfactory results, it is not ideal; human eyes cannot distinguish images by pixel difference but rather by perceptual quality. Generative Adversal Networks (or GANs) try to optimize the perceptual quality to produce images that are pleasant to observe. The most recognized GAN-related architectures are Super Resolution Generative Adversarial Network (SRGAN) which uses a GAN-based architecture to generate visually pleasing images. It uses the SRResnet network architecture as a backend and employs a multi-task loss to refine the results. Its loss is composed of Mean Square Error (MSE) loss, Perceptual similarity loss, and Adversarial loss (loss functions that will be reviewed deeper in the following chapter) (LEDIG et al., 2017). Subsequently, Enhanced Super Resolution Generative Adversarial Network (ESRGAN) is presented. a scheme improves on top of SRGAN by adding a relativistic discriminator. The advantage is that the network is trained not only to tell which image is true or fake but also to make real images look less real compared to the generated images, thus helping to fool the discriminator. Batch normalization in SRGAN is also removed, and Dense Blocks (inspired by DenseNet) are used for better information flow. These Dense Blocks are called Residual in Residual Dense Block (RRDB). (WANG et al., 2018b).

1.2 MOTIVATION

In recent years, SISR deep learning techniques have achieved remarkable improvements in recovering a HR image from an observed LR input. Nevertheless, these proposed methods adopt a pre-defined degradation process (e.g., bicubic downsampling) from an HR image to an LR one and fail in many real-world scenarios since their models are usually trained using a pre-defined degradation process from HR ground truth images to LR ones. To address this issue, new architectures have been proposed focusing on adopting more complicated degradation models to emulate real-world degradations achieving prominent performance but still limited to certain kinds of inputs and dropping considerably in other cases. The reason for this is that they still make some assumptions about the degradation types related to the input LR and inevitably produce much less pleasing results for input images with unknown degradations.

This is how new optimized deep learning models have been presented to tackle Blind Super Resolution (BSR) in recent years. Nevertheless, these proposed methods tend to fail in many real-world scenarios, their performance is usually limited to specific inputs and drops considerably in other cases. The main reason is that they still make some assumptions about the degradation types related to the input LR. *"While research on model-based blind single image super-resolution SISR has achieved tremendous success recently, most do not consider image degradation sufficiently. They assume image noise obeys an independent and identically distributed Gaussian or Laplacian distribution, which largely underestimates the complexity of real noise. Previous commonly-used kernel priors (e.g., normalization, sparsity) are not effective enough to guarantee a rational kernel solution and thus degenerates the performance of subsequent SISR task."* (YUE et al., 2022). That is why it is important to continue researching new strategies or algorithms that give significant solutions to the BSR problem. However, the tuning process is more empirical than theoretical; thus, most new proposals are essential modifications to previous SOTA models that work reasonably well against BSR. Some of these modifications consist of adding, or reducing the number of layers, convolutional blocks, optimizing parameters, hyperparameters, or loss functions. Consequently, considerable research has achieved significant progress in this field, such as kernel estimation, representation learning, zero-shot learning, meta-learning, optimization method, real-world dataset, and unsupervised methods (ZHANG et al., 2022). Despite all these efforts, down-sampling with blur degradation is still an overly simple simulation considering there exist many other degradation types in the real world.

Recently a new approach has emerged, presenting the implementation of U-Nets as components in GANs. Several proposals have been adopting this idea obtaining significant improvements in different computer vision solutions. Therefore, this project aims to contribute to the SOTA SISR focused on solving LR images that come from the real world since these have an unknown kernel distribution. To achieve this, it is presented a UR-SRGAN structure for BSR tasks, applying a technique that has not been widely employed in SISR: a U-Net architecture as a discriminator of the GAN network. Adding this structural change will encourage the discriminator to focus more on semantic and structural changes between real and fake images and to attend less to domain-preserving perturbations. In addition, the loss function of the generator was modified by adding the LPIPS loss function for the perceptual loss and a per-pixel consistency regularization technique based on the CutMix data augmentation. The proposed model will be trained using the different SR datasets employing a degradation framework for real-world images by estimating blur kernels and actual noise distributions to obtain more realistic LR samples. After all these modifications, it is figured to obtain a model that surpasses other more recent SOTA GAN models in the area of BSR. This will be confirmed and evaluated by performing a benchmark comparing all the metrics on a test dataset. The commonly-used evaluation metrics for image restoration, PSNR, SSIM, LPIPS, and NIQE, will be used for this evaluation.

1.3 OBJECTIVE

The overall objective of this work is to model and implement a GAN proposal oriented to SR for real-world images, namely UR-SRGAN (U-Net Real Super Resolution GAN). To achieve this, three important modifications will be made. The first consists of implementing a U-Net structure discriminator together to consider the global and local context of an input image. Furthermore, the addition of LPIPS and Feature Matching loss functions is proposed to obtain improvements in the perception of the image, and a pre-processing stage will be performed to retrieve low-resolution images that simulate an image with unknown degradation using kernel estimation and noise injection. Subsequently, training this UR-SRGAN architecture and comparing the results with other SOTA SRGAN models focused on restoring LR images with unknown and complex degradations at x4 scale and verifying the performance according to the PSNR, SSIM, LPIPS, and NIQE metrics.

1.4 THESIS STRUCTURE

In addition to the Introduction chapter, this thesis is divided into five supplementary chapters:

- Chapter 2: In this chapter, the topic of SISR will be developed. BISR and the relevant category for this work is Explicit Degradations Modeling. Later, more details will be given about the metrics that will be used to evaluate the results of the experiments of this work: PSNR, SSIM, LPIPS, and NIQE. Finally, a more detailed explanation of the U-Net network is presented and how it has been applied in other GAN models as a generator or discriminator, presenting important improvements in the results of these models recently related to computer vision.

- Chapter 3: This chapter describes a brief introduction to the GAN architecture and how it is composed by the generator and discriminator modules. Following, how it has been included into SR tasks and lastly, presenting the three most relevant SRGAN architectures so far focused on Real World Super Resolution (RWSR) tasks. This will help to understand how these solutions tackle the BSR problem and how was their implementation, datasets, training, and validation methods.

- Chapter 4: This chapter will be dedicated to further explaining the architecture of UR-SRGAN. Also, it describes the U-Net discriminator strategy and how it focuses more on semantic and structural changes between real and fake images. Furthermore, the justification for using this approach in the original ESRGAN architecture as a discriminator module. Later on, the second part will be to review the literature on a loss function proposal using LPIPS instead of the traditional perceptual loss based on VVG base architecture.

- Chapter 5: In this chapter, experiments will be carried out to confirm the effectiveness of the new UR-SRGAN architecture. First of all, this model will be trained together with the other three SOTA models (RealSR, Real-ESRGAN, A-ESRGAN) to verify that it satisfies the aim of generating HR images with a scale four times greater than the input image LR and achieving similar or better results than the counterpart models. All the models will be trained using the three previously mentioned image datasets (DIVerse 2K resolution high quality images (DIV2K), DF2K, DF2K+Outdoor Scenes (OST)) to see how the models behave with a greater number of images or with different types and characteristics. Finally, when obtaining the weights of all the trained models, an evaluation will be made in two phases. First, in the DIV2K validation dataset, which is used more frequently in competitions and recent research in the field of super-resolution, and later in the DPED validation dataset, which has real-world images obtained through different types of data. of phones and cameras. In the end, a global comparison will be made between all the models using the PSNR, SSIM, and LPIPS metrics as references for the images that have a GT reference and the NIQE metric for those that do not have said GT reference (real-world images).

- Chapter 6: In this chapter, the conclusions of the thesis will be detailed, including the limitations and experience during the experiments and future works into SISR concerning BISR.

2 BACKGROUND

2.1 INTRODUCTION TO SINGLE IMAGE SUPER-RESOLUTION

SISR aims to recover HR images from their LR counterparts. SISR is a fundamental problem in the community of computer vision and can be applied in many image analysis tasks, including surveillance and satellite images, as it was previously mentioned. SR is a widely known ill-posed problem since each LR input may have multiple HR solutions. With the development of deep learning, several deep SISR methods have been proposed and have largely boosted the performance of SR (MA et al., 2021).

In this section, it is introduced a more elaborate explanation of the mathematical formulations of the SISR schema. In particular (LIU et al., 2022) explains how SISR refers to the labor of reconstructing an HR image from a given LR input, especially the high-frequency contents in HR. The underlying degradation process from HR to LR can be generally expressed with Equation 2.1:

$$y = f(x; s), \tag{2.1}$$

where x and y denote HR image and LR image respectively, f is the degradation function with a scale factor of s. Thus, the SR problem is comparable with modeling and solving the inverse function f^{-1}. In the instance of non-blind SR, f is usually presumed to be bicubic downsampling, as shown in Equation 2.2:

$$y = x \downarrow_s^{bic}, \tag{2.2}$$

Alternatively, it can also be considered as the combination of downsampling and a fixed Gaussian blur with kernel k_g, as shown in Equation 2.3:

$$y = (x \otimes k_g) \downarrow_s, \tag{2.3}$$

where $\otimes$ indicates a convolutional operation. Under this premise, the SR model is only capable of handling LR inputs with this specific type of degradation. For other LR images with different degradation kinds, the inconsistency between the SR model and intrinsic degradation of inputs might produce various artifacts in SR results.

Consequently, the topic of BSR for unknown degradation is proposed to bridge this gap. So far, two different ways of modeling the degradation process for BSR: explicit modeling based on an extension of Equation 2.3, and implicit modeling through inherent distribution within the external dataset. To be specific, explicit modeling usually employs a so-called classical degradation model, which is a more general form of Equation 2.4:

$$y = (x \otimes k) \downarrow_s +n, \tag{2.4}$$

where the SR blur kernel k and additive noise n are two main factors involved in the degradation process, and parameters related to these two factors will be unknown for an arbitrary LR input.

2.2 BLIND SINGLE-IMAGE SUPER-RESOLUTION

SISR has long been a fundamental problem in low-level vision, focusing on the recovery of a HR image from an observed LR input. In recent years, the community has performed remarkable progress in this field, especially with the prosperous deep learning techniques. However, most existing methods assume a pre-defined degradation process from an HR image to an LR one, which can hardly hold for real-world images with complex degradation types. In the direction of filling this gap, more attention has been paid in recent years to approaches for unknown degradations in real-world applications, namely BSR (LIU et al., 2022). Despite several impressive improvements, these proposed solutions fail in many real-world scenarios, as their performance is usually limited to certain kinds of inputs and will drop considerably in other cases. The main reason is that they still make some assumptions about the degradation types related to the input LR.

The ways of degradation modeling in BISR can be simply divided into two categories: explicit degradation modeling methods and implicit degradation modeling methods. Among them, explicit degradation modeling methods can be further divided into two categories according to whether they use kernel estimation technology.

2.2.1 Explicit Degradation Modeling

This section covers recently proposed BSR methods with explicit modeling of the degradation process, usually based on the classical degradation model shown by Equation 2.4. Besides, these approaches can be further classified into two sub-classes according to whether they employ an external dataset or rely on a single input image to solve the SR problem.

2.2.1.1 Classical Degradation Model with an External Dataset

This kind of approach utilizes an external dataset to train an SR model well adapted to variant SR blur kernels k and noises n. Typically, the SR model is parameterized with a convolutional neural network CNN, and an estimation on k or n for a specific LR image is used as conditional input to the SR model for feature adaptation purposes. After the training, the model can produce satisfactory results for LR inputs with degradation types covered in the training dataset. According to whether a certain approach includes degradation estimation in its proposed framework, in this category, there are two approaches to take into consideration:

- Image-specific adaptation without kernel estimation: Receives estimated degradation information as additional inputs and is focused on how to utilize the estimation input for image-specific adaptation.

- Image-specific adaptation with kernel estimation: Provides special attention to kernel estimation along with the SR process.

2.2.1.1.1 Image-specific Adaptation without Kernel Estimation

In this section, it will be denoted some relevant approaches for SR methods that perform without kernel estimation. For instance, SR for Super Resolution for Multiple Degradations (SRMD) proposes to directly concatenate an LR input image with its degradation map as a unified input to the SR model, thus allowing feature adaptation according to the specific degradation and covering multiple degradation types in a single model (ZHANG; ZUO; ZHANG, 2018). This strategy can be easily extended to non-uniform maps for spatially variant degradations. The SR reconstruction network of SRMD is similar to those commonly adopted in non-blind SR.

Following SRMD, more architectures were labored like Unified Dynamic Convolutional Network for Variational Degradations (UDVD), which uses the degradation map as an additional input for SR reconstruction. It makes one step forward by employing per-pixel dynamic convolution to more effectively deal with variational degradations across images (XU et al., 2020). Afterward, Deep Plug-and-play Super Resolution (DPSR) incorporates an SR network into a MAP-based iterative optimization scheme and proposes a principled formulation and framework by extending bicubic degradation-based deep SISR with the help of a plug-and-play framework to handle LR images with arbitrary blur kernels (ZHANG; ZUO; ZHANG, 2019).

USRNet also adopts the MAP framework but is based on the original degradation model in Equation 2.4 which super-resolves an LR image blurred by kernel k and denoises an glshr image with a virtual noise level μ (ZHANG; GOOL; TIMOFTE, 2020). It enhances the solution framework by unfolding the iterative optimization process of DPSR into an end-to-end trainable network with the iterative scheme. Some examples of SR frameworks without kernel estimation were presented to be more familiar with this concept. Nevertheless, as (LIU et al., 2022) mentioned, these architectures have limitations. They all rely on an additional input of degradation estimation, especially the SR kernel k. However, estimating the correct kernel from an arbitrary LR image is not an easy task, and an inaccurate estimation input will cause kernel mismatch and greatly compromise the SR performance. In this manner, the next part will introduce another kind of approach, which incorporate kernel estimation into the SR framework for more robust performance and on which this work focuses more since the main architecture, its modifications, and experiments are within this category. Therefore more emphasis will be given to explaining more examples and details about these solutions for SR.

2.2.1.1.2 Image-specific Adaptation with Kernel Estimation

In this segment, SR architectures based on kernel estimation will be further developed. Primarily, it is essential to denote the Iterative Kernel Correction (IKC) whose main objective is to correct the kernel estimate iteratively to approach a suitable result progressively. What stands out much more about this method is taking advantage of intermediate SR results since artifacts into an SR image caused by kernel mismatch tend to have regular patterns (GU et al., 2019). Another example is the Deep Alternate Network (DAN) (HUANG et al., 2020b). This strategy improves the IKC framework much more. It unifies the corrector and the SR network into one trainable end-to-end instead of training each subnet separately, as is the case with IKC. This joint formation proposal can make the two networks more compatible. In addition, the corrector manipulates the original LR input for kernel calculation depending on an intermediate SR result, which is favorable for more stable kernel estimation performance. (LUO et al., 2020). The approach of making use of SR artifacts for kernel estimation is also employed in variant BISR (VBSR), which trains a kernel discriminator to estimate the error map of an SR output instead of the kernel itself. and finds the optimal kernel by minimizing the output error SR during the inference stage. However, an iterative scheme like this consumes more inference time and requires human intervention to select the optimal number of iterations (CORNILLèRE et al., 2019).

To address this issue, some recent works propose non-iterative frameworks by introducing more accurate degradation estimation or more efficient feature adaptation strategies. Unsupervised degradation representation learning for BSR (DRL-DASR) aims to estimate the degradation details with a trainable encoder in the latent feature space, and the degradation encoder is trained in an unsupervised manner (WANG et al., 2021a). Kernel-oriented local adaptive tuning (KOALAnet) also uses a similar dynamic kernel strategy that adapts the SR network to specific degradation. Further, it extends the non-iterative framework to spatially variable impairment with a reduction network sampling for local kernel estimation (KIM; SIM; KIM, 2021). Eventually, the adaptive modulation network with reinforcement learning (AMNet-RL) proposes a modified version of the adaptive instance norm, known as AdaIN, to add the kernel estimation in the SR network. It was also a very advanced proposal in optimizing the BSR model with indistinguishable perceptual metrics under reinforcement learning framework (HUI et al., 2021).

However, there are also other approaches proposing to learn a BSR model by merely covering more degradations in the training dataset, especially more realistic kernels estimated from real images, which will be of more interest to describe for this project. For instance, Kernel Modeling Super Resolution (KMSR) constructs a large kernel pool with data distribution learning based on some realistic SR kernels estimated from real LR images. Kernels from this pool are then used to synthesize HR-LR training pairs according to the classical degradation model, and the training process just follows a non-blind setting with supervised learning. (ZHOU; SüSSTRUNK, 2019). In other words, the SR model will be implicitly granted more capacity for kernel estimation in the training process, thus avoiding explicit kernel estimation in the framework. However, such a direct way may not lead to top performance. That is why a homogeneous strategy is employed in RealSR (JI et al., 2020a), and Real-ESRGAN (REN et al., 2020). Its variation uses an attention module into de U-Net discriminator A-ESRGAN (WEI et al., 2021a) to build a more generic training dataset with more realistic kernels. These last three architectures will be the base to perform the experiments to compare both author's results and the modifications that are proposed in the current thesis and verify whether this project shows a better performance than the original ones according to SR metrics.

2.3 IMAGE QUALITY ASSESSMENT

Image quality is the characteristic of an image that measures the perceived image degradation (typically compared to an ideal or perfect image). In general, Image Quality Assessment (IQA) methods include subjective methods based on human perceptions. For example, how realistic the image looks or objective computational methods. The outcome is more aligned with our needs but is often time-consuming and expensive. Thus the latter is currently the mainstream. Nonetheless, these methods are not necessarily consistent with each other because objective methods are often unable to capture precisely the human visual perception, which might lead to a large difference in IQA results.

Furthermore, the objective IQA methods are further categorized into three types: full-reference methods performing assessment using reference images, reduced-reference methods based on comparisons of extracted features, and no-reference methods without any reference images. The most commonly used IQA methods will be introduced, covering both subjective and objective methods (WANG; CHEN; HOI, 2020).

2.3.1 Peak Signal-to-noise Ratio

The PSNR is one of the most popular reconstruction quality measurements of lossy transformation. For ISR, PSNR is defined via the maximum pixel value denoted as L and the mean squared error (MSE) between images. Given the GT image I with N pixels and the reconstruction $\hat{I}$, the PSNR between I and $\hat{I}$ are defined as follows in Equation 2.5:

$$PSNR = 10 \cdot \log_{10}(\frac{L^2}{\frac{1}{N}\sum_{i=1}^{N}(I(i) - \hat{I}(i))^2}), \tag{2.5}$$

where L equals 255 in general cases using 8-bit representations. Because the PSNR is only related to the pixel-level MSE only cares about the differences between corresponding pixels instead of visual perception, it frequently leads to low performance in representing the reconstruction quality in natural scenes, where usually human perception is more valued. However, due to the necessity to compare with literary works and the lack of entirely accurate perceptual metrics, PSNR is still currently the most widely used evaluation criteria for SR models.

2.3.2 Structural Similarity

Considering that the human visual system is highly adapted to extract image structures, the SSIM metric is proposed for measuring the structural similarity between images based on independent comparisons in terms of luminance, contrast, and structures (WANG et al., 2004). For an image I with N pixels, the luminance μ_I and contrast σ_I are estimated as the mean and standard P deviation of N_1 the image intensity, respectively, as it is illustrated at this point: $\mu_I = \frac{1}{N}\sum_{i-1}^{N} I(i)$ and $\sigma_I = (\frac{1}{N-1}\sum_{i-1}^{N}(I(i) - \mu_I)^2)^{\frac{1}{2}}$, where $I(i)$ represents the intensity of the i^th pixel of the image I. And the comparisons on luminance and contrast, denoted as $C_l(I, \hat{I})$ and $C_c(I, \hat{I})$ respectively, are given by Equations 2.6 and 2.7 :

$$C_l(I, \hat{I}) = \frac{2\mu_I\mu_{\hat{I}} + C_1}{\mu_I^2 + \mu_{\hat{I}}^2 + C_1}, \tag{2.6}$$

$$C_c(I, \hat{I}) = \frac{2\sigma_I\sigma_{\hat{I}} + C_2}{\sigma_I^2 + \sigma_{\hat{I}}^2 + C_2}, \tag{2.7}$$

where $C_1 = (k_1 L)^2$ and $C_2 = (k_2 L)^2$ are constants for avoiding instability, $k_1 \ll 1$ and $k_2 \ll 1$. In addition, the image structure is represented by the normalized pixel values: $(I - \mu_I)/\sigma_I$, whose correlations measure the structural similarity, equivalent to the correlation coefficient between I and $\hat{I}$. Thus structure comparison function $C_s(I, \hat{I})$ is defined as follows by Equations 2.8 and 2.9:

$$\sigma_{I,\hat{I}} = \frac{1}{N-1}\sum_{i=1}^{N}(I(i) - \mu_I)(\hat{I}(i) - \mu_{\hat{I}}), \tag{2.8}$$

$$C_s = \frac{\sigma_{I,\hat{I}} + C_3}{\sigma_I\sigma_{\hat{I}} + C_3}, \tag{2.9}$$

where $\sigma_{I,\hat{I}}$ is the covariance between I and $\hat{I}$, and C_3 is a constant for stability. At last, the SSIM is given by Equation 2.10:

$$SSIM_{I,\hat{I}} = [C_l(I,\hat{I})]^{\alpha}[C_l(I,\hat{I})]^{\beta}[C_l(I,\hat{I})]^{\gamma}, \tag{2.10}$$

where α, β, γ are control parameters for adjusting the relative importance. Since the SSIM evaluates the reconstruction quality from the perspective of the human visual system, it better finds the requirements of perceptual assessment and is also widely used.

2.3.3 Learning-based Perceptual Quality

To better assess the image perceptual quality while reducing manual intervention, researchers try to assess the perceptual quality by learning on large datasets. (MA et al., 2017) and (TALEBI; MILANFAR, 2018) propose no-reference Mult Adds (MA) and Neural Image Assessment (NIMA), respectively, which are learned from visual perceptual scores and directly predict the quality scores without GT images. In contrast, (KIM; LEE, 2017) proposes DeepQA, which predicts the visual similarity of images by training on triplets of distorted images, objective error maps, and subjective scores. And (ZHANG et al., 2018) collect a large-scale perceptual similarity dataset, evaluate the LPIPS according to the difference in deep features by trained deep networks, and show that the deep features learned by CNN model perceptual similarity much better than measures without CNN. Although these methods exhibit better performance in capturing human visual perception, the objective IQA methods such as PSNR and SSIM are still the mainstream currently. The last-mentioned LPIPS will be considered a perception metric in the experiments executed for the changes proposed in this hypothesis.

2.3.4 Natural Image Quality Evaluator

Proposed in (MITTAL; SOUNDARARAJAN; BOVIK, 2012), NIQE (Natural Image Quality Evaluator) is the first proposed OU-DU-NR-IQA (Image Quality Assessment) metric. OU stands for Opinion-Unaware, as opposed to Opinion-aware IQA; it does not require the human subjective opinion score. DU stands for Distortion-unaware, as opposed to Distortion-aware IQA, DU does not require prior knowledge of how an image is downscaled. NR stands for no-reference, which means the evaluation is based on the evaluated image alone, without referring to the ground truth pairing image.

NIQE uses the measurable evictions from statistical regularities observed in natural images. First of all, it computes a spatial domain NSS (Natural scene statistic) model by computing the local mean removal and divisive normalization, as shown in Equation 2.11 :

$$I(i,j) = \frac{I(i,j) - \mu(i,j)}{\sigma(i,j) + 1} \tag{2.11}$$

Where $\mu(i,j)$ and $\sigma(i,j)$ are local mean and contrast at location (i,j), computed with a 2D circularly-symmetric Gaussian weighting function with deviation equal to three. It is observed that for natural images, the coefficients of NSS features obtained from the equation above can fit well into a Gaussian distribution. In contrast, the distorted images fail to do so. Thus, a multivariate Gaussian (MVG) fit of the NSS features extracted from the natural image corpus is computed. Finally, NIQE is computed as the distance between the MVG fit of the test image and the MVG fit of the natural image corpus. The distance, $\upsilon_1, \upsilon_2, \Sigma_1, \Sigma_2$ stands for the mean and variance matrices of the natural MVG model and the distorted images MVG model, better explained as follows by Equation 2.12:

$$D(\upsilon_1, \upsilon_2, \Sigma_1, \Sigma_2) = \sqrt{((\upsilon_1 - \upsilon_2)^T (\frac{\Sigma_1 + \Sigma_2}{2})^{-1} (\upsilon_1 - \upsilon_2))} \tag{2.12}$$

As shown in 2.12, NIQE can predict the image quality with little prior knowledge of the GT image or its distortions. And is claimed by authors of NIQE through experiment comparisons to perform equally or better than other IQA such as SSIM.

2.4 DATASETS FOR SUPER-RESOLUTION

There are a large diversity of datasets available for image SR, which considerably differ in image amounts, quality, resolution, and diversity. Some of them provide LR-HR image pairs. In contrast, others only provide HR images, which case the LR images are generally obtained by bicubic interpolation with anti-aliasing. In Table 1 it is listed several image datasets commonly used by the SR community and in official competitions. It indicates their amounts of HR images, average resolution, average numbers of pixels, image formats, and category keywords (WANG; CHEN; HOI, 2020). Furthermore, combining multiple datasets for training is also popular. For example, (KIM; LEE; LEE, 2016b) a model that exploits contextual information over large image regions in an efficient way by cascading small filters many times in a deep network structure, and (LAI et al., 2017) the Laplacian Pyramid Super-Resolution Network (LapSRN) to progressively reconstruct the sub-band residuals of high-resolution images. Both of these models combine T91 and BSDS300 datasets for training. It is also frequently seen in the combination of DIV2K (TIMOFTE et al., 2017) and Flickr2K (AGUSTSSON; TIMOFTE, 2017), better known as the DF2K dataset, which will be further detailed and also employed during all the experiments, including also the OST dataset. In addition, some image datasets are acquired by common devices such as cell phones or cameras that are available for this type of image enhancement and glssr tasks. In the present work, the DPED dataset will be used to evaluate the generation of synthetic images from the UR-SRGAN architecture. Table 1, displays more information about the datasets that are most frequently used in the SR research area. In each column, we can see much more details about the number of images that each dataset has, the average of their resolutions, what is their format, and keywords that indicate the classes of images that each source includes.

Table 1 – List of public image datasets for SR benchmarks.

Dataset	Amount	Avg. Resolution	Avg. Pixels	Format	Category Keywords
BSDS300	300	(435, 367)	154, 401	JPG	animal, building, food, landscape, people, plant, etc.
BSDS500	500	(432, 370)	154, 401	JPG	animal, building, food, landscape, people, plant, etc.
DIV2K	1000	(1972, 1437)	2, 793, 250	PNG	environment, flora, fauna, handmade object, people, scenery, etc.
General-100	100	(435, 381)	181, 108	BMP	animal, daily necessity, food, people, plant, texture, etc.
L20	20	(3843, 2870)	11, 577, 492	PNG	animal, building, landscape, people, plant, etc.
Manga109	109	(826, 1169)	966, 011	PNG	manga volume
OutdoorScene	10624	(553, 440)	249, 593	PNG	animal, building, grass, mountain, plant, sky, water
PIRM	200	(617, 482)	292, 021	PNG	environments, flora, natural scenery, objects, people, etc.
Set5	5	(313, 336)	113, 491	PNG	baby, bird, butterfly, head, woman
Set14	14	(492, 446)	230, 203	PNG	humans, animals, insects, flowers, vegetables, comic, slides, etc.
T91	91	(264, 204)	58, 853	PNG	car, flower, fruit, human face, etc.
Urban100	100	(984, 797)	774, 314	PNG	architecture, city, structure, urban, etc.

Source: (WANG; CHEN; HOI, 2020)

2.5 U-NET GENERATIVE ADVERSARIAL NETWORK

2.5.1 U-Net Convolutional Neural Network

Convolutional neural network (CNN), particularly the U-Net, is a powerful method for medical image segmentation. By this time, U-Net has demonstrated SOTA performance not only in many complex medical image segmentation tasks but also in numerous variations. As U-Net's potential is still increasing, these variations developments have been developed further, approaching several areas of computer vision. Therefore, it is presented how this U-Net architecture started being designed for medical segmentation and eventually gained more popularity in other most recent proposals, including GAN solutions, which present substantial improvements to its previous implementation and demonstrate that it might also have notable results in the proposal to apply SR to images with unknown degradation of the real world.

(RONNEBERGER; FISCHER; BROX, 2015) presented U-Net fully connected neural network in 2015 and applied it to medical image segmentation. Due to the problem of small medical image data samples, U-Net adopts a symmetrical U-shaped structure to extract the feature information in the image samples. The network structure is divided into the downsampling shrinking process and the upsampling expansion process. There are four downsampling operations, each doubles the number of feature channels of the image by increasing the number of convolution kernels. After convolution, global average pooling (GAP) is used to reduce the size of the feature map to reduce the difficulty of network training. This incremental increase in the number of feature channels by layer-by-layer convolution not only reduces the burden of training a fully convolutional network but also can fully extract the beneficial part of the image information.

Upsampling is achieved by deconvolution. During the upsampling expansion of the feature map, the number of convolution kernels is halved layer by layer, and the feature map size is recovered layer by layer using deconvolution. In the U-shaped network, the feature information of systolic and dilated paths in the same layer are fused by skip connection before feature extraction through the convolution layer. The location information extracted from the contraction path in the U-shaped structure is combined with the high-level feature information extracted from the expansion path, which provides attention to the network to a certain extent. In addition, the image details lost in the downsampling process can also be compensated accordingly through the symmetric network structure, reducing the loss of image information in continuous convolution.

Many U-Net-based models have been proposed; for instance, (KERFOOT et al., 2018) used a U-Net convolutional neural network architecture built from residual units to segment the left ventricle. UNet++, proposed by (ZHOU et al., 2018), introduced nested and dense skip connections to reduce the semantic gap between the encoder and decoder. Although reasonable performance can be achieved, the nested network structure is too complex and cannot examine enough information from the full scale. (WENG et al., 2019) proposed NAS-UNet, using three types of primitive operation sets and search space to automatically find two cell architectures, DownSC and UpSC, for medical image segmentation, which attains better performance and uses much fewer parameters than standard U-Net. (HUANG et al., 2020a) uses comprehensive skip connections to aggregate feature maps of all scales at each feature fusion, using full-scale feature information. Reasonable results can be obtained using UNet 3+ but with fewer parameters than U-Net. (LOU; GUAN; LOEW, 2021) analyzed the classical U-Net and the recent MultiResUNet (IBTEHAZ; RAHMAN, 2020) architecture and then designed the Dual-Channel CNN block to provide more effective features with fewer parameters. However, U-Net has not only been used in the mentioned approaches but has also been implemented recently in GANs, as shown in the next topic.

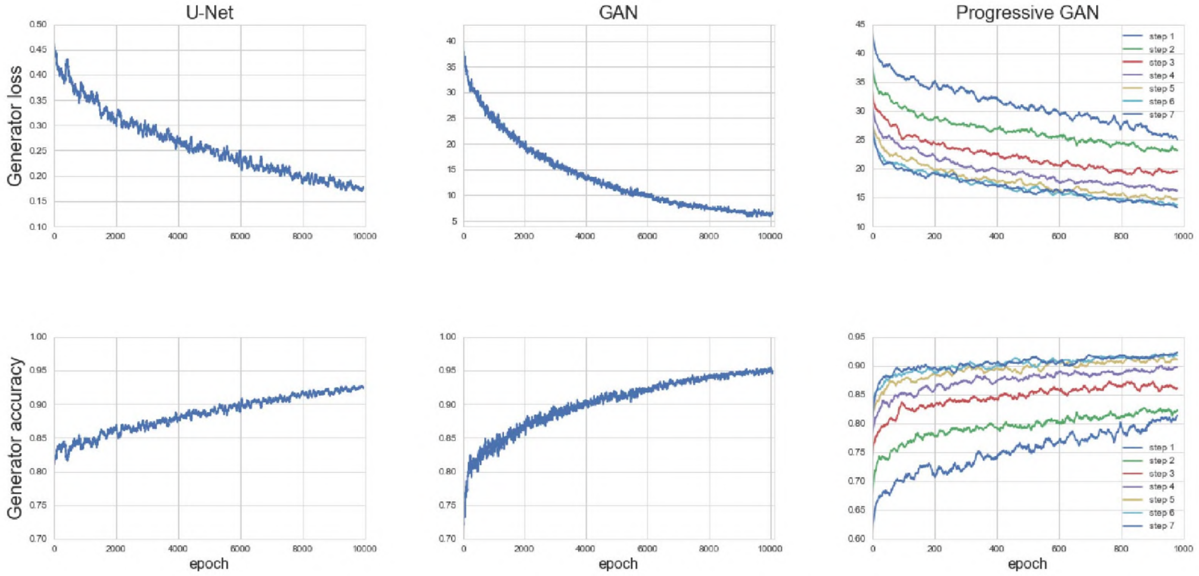
2.5.2 U-Net Into GAN-based Architectures

Since its introduction in 2014, generative adversarial network (GAN) has achieved remarkable success in generative image modeling and has shown outstanding performances in numerous applications. The architecture of the generative adversarial network integrates two competing networks, a generative network, and a discriminative network, into one framework. The generator is to map given data to synthetic samples, and the discriminator is to differentiate the generated synthetic samples from the real samples. The two networks are trained sequentially and iteratively in a competing manner to boost the performance of the other. The final goal is to generate synthetic samples that cannot be differentiated from real samples.

The employment of the U-Net architecture in the GAN strategy is a novel proposal taken by several investigations and top-notch deep learning solutions for computer vision. In particular, the addition of a U-Net in GAN models can be used to act as a generator for an end-to-end network and introduce extra judgment with the discriminator to help the generator find the optimal solutions. It can also act as a discriminator, U-Net discriminators focus more on semantic and structural changes between real and fake, whereas, in GAN, two neural networks compete with each other to become more accurate in their predictions by creating their training data and automatically discovering and learning regularities to generate new samples that plausibly could have been drawn from the original dataset by framing the problem as a supervised learning problem with two sub-models. The generator generates new examples, and the discriminator tries to classify these examples as either real (from the domain) or fake (generated).

(COLLIER et al., 2018) evaluate the efficacy of progressive training of a generative adversarial network (GAN) for rooftop segmentation using multi-spectral satellite images. This GAN network consists of a generator and a discriminator linked through an adversarial training algorithm. The generator learns to generate mappings from input to target, and the discriminator learns to evaluate them. Feedback from the discriminator enables the generator to produce highly realistic outputs. The U-Net architecture, composed of a convolutional neural network consisting of an encoder-decoder, was employed as the generator. Mirrored layers in U-Net contain skip connections that allow structural information to be preserved when decoding from the learned latent encoding. Also, progressive growth of the generator and discriminator was applied. Therefore, in this transfer learning process, deep networks are trained to learn increasingly complex features (YOSINSKI et al., 2014). The accuracy of rooftop classification is assessed, and results are compared with those of a traditionally trained generative model and a non-generative U-Net. Figure 4 shows the generator loss and accuracy over training epochs for a single U-Net network, GAN, and Progressive GAN. The progressive GAN converge to a better performance with each progressive step until some ceiling is reached and improves the definition of individual buildings compared to their counterparts. Consequently, this demonstrates an improvement in semantic segmentation performance by GANs using progressive growing using a U-Net architecture embedded as the GAN generator.

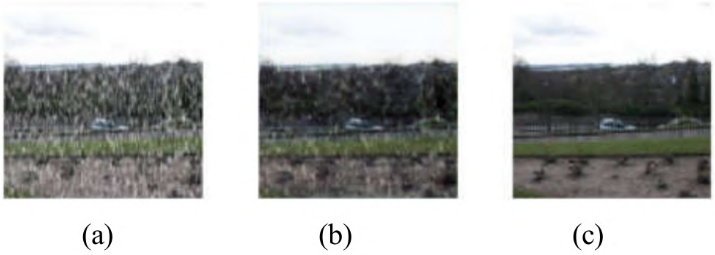
Figure 4 – Generator loss and accuracy over training epochs for U-Net, GAN, and Progressive GAN. For our proposed model, the progressive GAN, generator accuracy, and loss converge to an increasingly better performance with each progressive step until some ceiling is reached. The increasing resolution does not result in learning finer features.



Source: (COLLIER et al., 2018)

(RAMWALA; PAUNWALA; PAUNWALA, 2019) seeks to leverage the generative modeling capabilities of Generative Adversarial Networks by utilizing particular architectures for the generator and discriminator. The generator network is a Fully Convolutional U-Net architecture, and the discriminator is a standard binary cross-entropy classifier intended to classify whether the predicted de-rained image of the generator matches the real high-resolution image or not. The generator network has a U-Net architecture (WANG et al., 2018) divided into three segments; a contracting or downsampling path, a bottleneck part, and an expanding or upsampling path. This U-Net generator has a symmetric architecture. The upsampling and the downsampling segments have skip connections between them that utilize a concatenation operator, which gives local details to the global data during upsampling. Due to its symmetry, it has many feature maps in the expanding path, which provide information transfer. Metrics used for quantitative comparison include PSNR (Peak Signal to Noise Ratio), SSIM (Structural Similarity Index Measure), UQI (Universal image Quality Index), and VIF (Variance Inflation Factor). Their proposed architecture was trained for 30 epochs, which indicates reduced computational complexity as compared to novel approaches. Figure 5 displays the results of the single U-Net architecture and the output of the U-Net-based GAN for the given input rainy image.

Figure 5 – a) Rain-Degraded Image b) U-Net c) Proposed U-Net based GAN method.

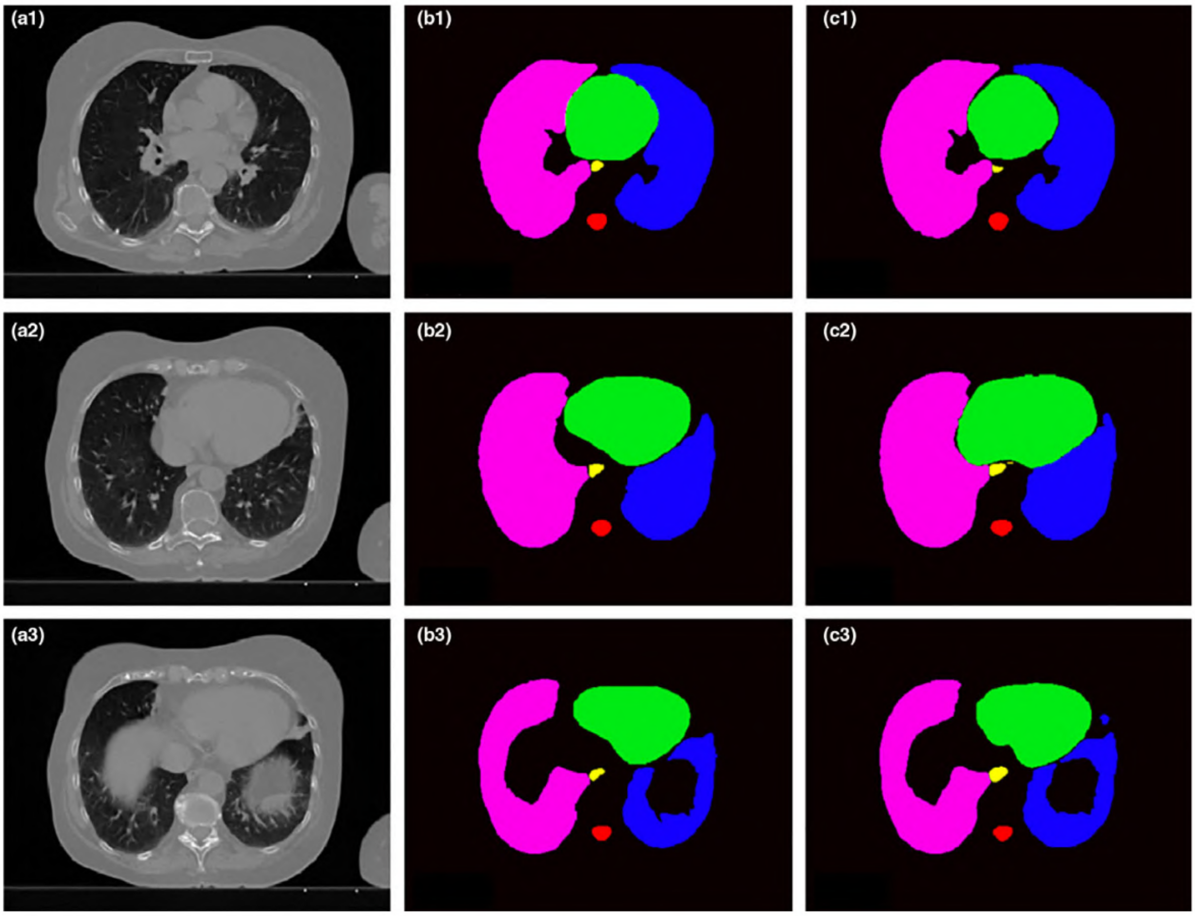


Source: (RAMWALA; PAUNWALA; PAUNWALA, 2019)

(DONG et al., 2019) propose an adversarial training strategy to train deep neural networks for segmenting multiple organs on thoracic CT images. The proposed design of adversarial networks, called U-Net-generative adversarial network (U-Net-GAN), jointly trains a set of U-Nets as generators and fully convolutional networks (FCNs) as discriminators. Specifically, the generator, composed of U-Net, produces an image segmentation map of multiple organs by an end-to-end mapping learned from CT images to multiorgan-segmented OARs (ECABERT et al., 2008). The discriminator, structured as an FCN (Fully Convolutional Network), discriminates between the ground truth and segmented OARs produced by the generator. The generator and discriminator compete against each other in an adversarial learning process to produce the optimal segmentation map of multiple organs. This segmentation technique was applied to delineate the left and right lungs, spinal cord, esophagus, and heart using 35 patients' chest CTs. Their novel deep learning-based approach with a GAN strategy to segment multiple OARs in the thorax using chest CT images demonstrated its feasibility and reliability, becoming a potentially valuable method for improving the efficiency of chest radiotherapy treatment planning. Figure 6 shows the 2D segmentation results on one patient using the proposed U-Net-GAN method. The proposed method segments bilateral lungs, heart, and spinal cord and successfully delineates the esophagus. The OARs obtained with their method show a great resemblance to the ground truth contours.

(HUANG et al., 2021) proposes a novel method to regularize better the Low-dose computed tomography (LDCT) denoising model in medical imaging, termed DUGAN, which leverages U-Net-based discriminators in the GANs framework not only to learn both global and local differences between the denoised and normal-dose images in both image and gradient domains but also focus on the global structure. This helped to alleviate the artifacts caused by photon starvation and enhance the edge of the denoised CT images in the image gradient domain. Their experiments demonstrated the effectiveness of their proposed method through visual comparison and quantitative comparison. The datasets were a simulated LDCT, and a real-world dataset from (YI; BABYN, 2018) that included 850 CT scans of a deceased piglet obtained by a GE scanner. In the generated images,

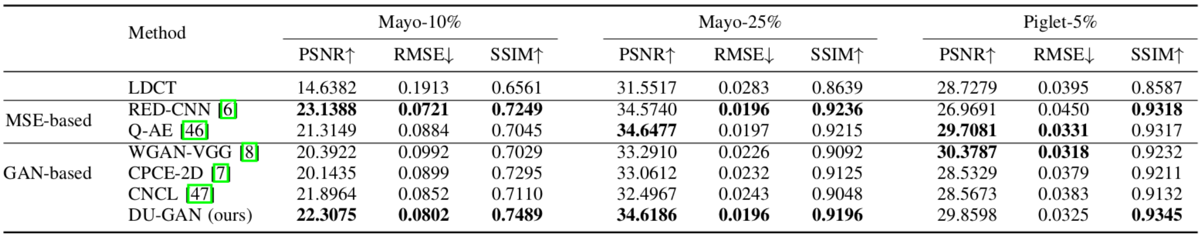
Figure 6 – (a) Three transverse computed tomography slices on one patient and the corresponding organ-at-risk contours obtained from (b) manual contouring (ground truth) and (c) the proposed method.



Source: (DONG et al., 2019)

the small structures' boundaries are consistently preserved with clear visual fidelity. This benefits from the added U-Net-based discriminator, which can provide feedback on global structures and local details to the generator, compared to the traditional classification discriminator used in WGAN-VGG and CPCE-2D with only structure information. Besides, the gradient domain branch can also encourage the denoising model to preserve edge information better. Figure 7 presents a table with the results of different methods. First, RED-CNN and Q-AE are MSE-based denoising methods as they are directly trained with solely MSE loss. Although they achieve better PSNR and RMSE results, the visual results in Figure 8 confirm that MSE-based methods produce over-smoothed results compared to the NDCT images, leading to loss of structural information and the over-smoothed denoising results lead to a lower SSIM score. Second, WGAN-VGG, CPCE-2D, and DU-GAN are GAN-based methods. Consequently, the results of DU-GAN preserve more structural details important for diagnosis, at the cost of compromising the quantitative metrics such as PSNR and RMSE.

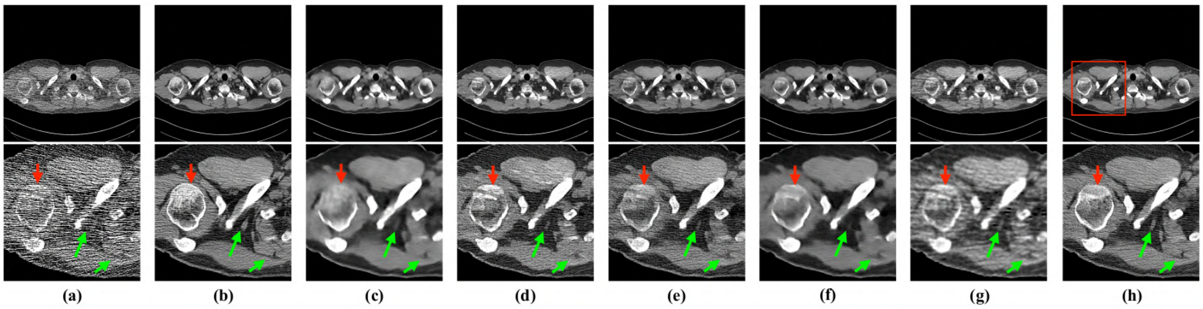
Figure 7 – Quantitative comparisons of different methods on the testing sets of two simulated datasets and one real-world dataset. The best results among MSE and GAN based methods are marked in bold.



	Method	Mayo-10%			Mayo-25%			Piglet-5%		
		PSNR↑	RMSE↓	SSIM↑	PSNR↑	RMSE↓	SSIM↑	PSNR↑	RMSE↓	SSIM↑
	LDCT	14.6382	0.1913	0.6561	31.5517	0.0283	0.8639	28.7279	0.0395	0.8587
MSE-based	RED-CNN [6]	**23.1388**	**0.0721**	**0.7249**	34.5740	**0.0196**	**0.9236**	26.9691	0.0450	**0.9318**
	Q-AE [46]	21.3149	0.0884	0.7045	**34.6477**	0.0197	0.9215	**29.7081**	**0.0331**	0.9317
GAN-based	WGAN-VGG [8]	20.3922	0.0992	0.7029	33.2910	0.0226	0.9092	**30.3787**	**0.0318**	0.9232
	CPCE-2D [7]	20.1435	0.0899	0.7295	33.0612	0.0232	0.9125	28.5329	0.0379	0.9211
	CNCL [47]	21.8964	0.0852	0.7110	32.4967	0.0243	0.9048	28.5673	0.0383	0.9132
	DU-GAN (ours)	**22.3075**	**0.0802**	**0.7489**	**34.6186**	**0.0196**	**0.9196**	29.8598	0.0325	**0.9345**

Source: (HUANG et al., 2021)

Figure 8 – Transverse neck CT images from the Mayo-10%: (a) LDCT; (b) NDCT; (c) RED-CNN; (d) WGAN-VGG; (e) CPCE-2D; (f) Q-AE; (g) CNCL; and (h) DU-GAN. Zoomed ROI of the red rectangle is shown in the second row. The red arrow indicates the bone area, while the green arrow indicates a small structure.



Source: (HUANG et al., 2021)

As demonstrated, the implementation of U-Net in GAN networks has significantly improved the different deep-learning solutions applied to image segmentation, enhancement, and de-raining. However, not much has been done in the field of super-resolution. These contributions inspire the current work to use the effective U-Net architecture in a GAN model, in this case, designing the discriminator into a U-shaped structure, which can provide per-pixel feedback to the generator and promote the generator to generate a more realistic high-resolution image. In the next chapter, these technologies and methods will be developed in more detail and applied to the problem in the context of obtaining high-resolution images through the proposal UR-SRGAN model of this work.

3 GENERATIVE ADVERSARIAL NETWORK ARCHITECTURE FOR REAL-WORLD SUPER RESOLUTION

3.1 INTRODUCTION

In this chapter, a brief introduction to the concept of Generative Adversarial Networks will be given and how the generator and discriminator strategy works. This will be essential to continue to the next part, where the application of GAN schemes in SR will be developed. Subsequently, the most recent techniques will be mentioned. The best results presented in official ISR publications and competitions, which are based on a GANs architecture, were employed as reference and inspiration to give way to the implementation of the new proposed architecture for this work and which will also be a guide to compare the results obtained in the experiments of this work concerning the models already presented in this section.

3.2 GENERATIVE ADVERSARIAL NETWORKS

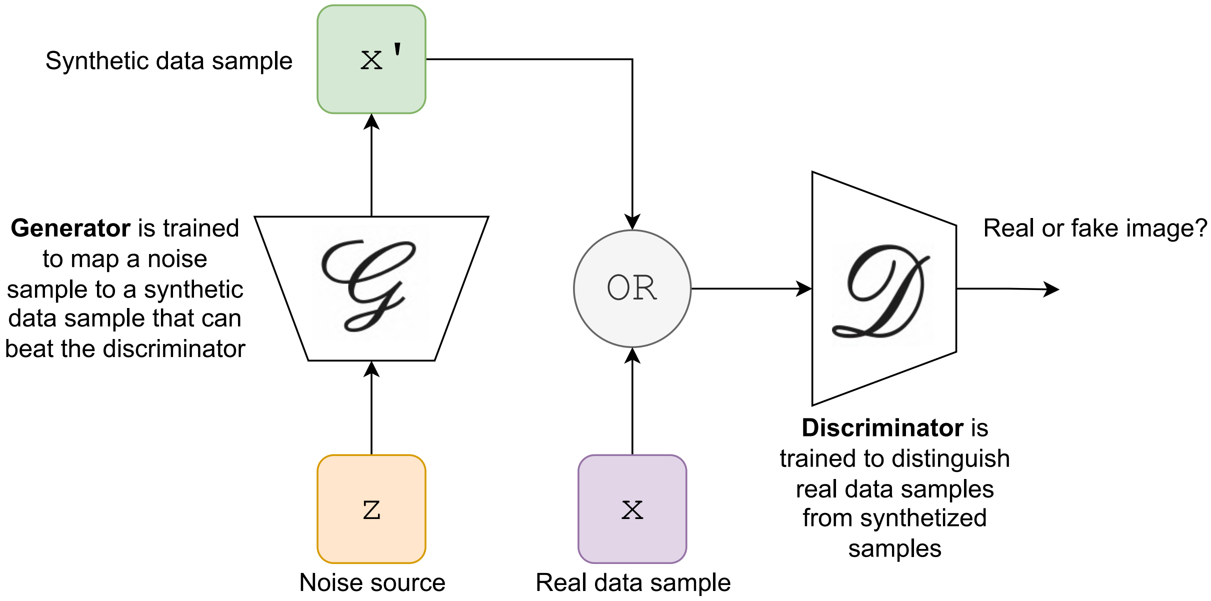
GAN are a developing technique for both semi-supervised and unsupervised learning. They offer a way to learn deep representations without broadly annotated training data. They accomplish this by deriving backpropagation signals during a competitive process involving a pair of networks. The representations that GANs can learn may be used in various applications, including image synthesis, classification, style transfer, semantic image editing, and ISR.

GANs are an optimal way of training a generative model by framing the problem as a supervised learning problem with two sub-models: the generator model, $\mathcal{G}$, which is trained to generate new image samples, and the discriminator model that tries to classify these generated images as either real (from the domain) or fake (generated). The two models are trained together in a zero-sum game, adversarial, until the discriminator model, $\mathcal{D}$, is deluded about half the time, meaning the generator model generates reasonable output images. Figure 9 shows how both generator and discriminator are trained simultaneously, and in competition with each other.

Essentially, the generator has no direct access to real images. The only way it can learn is through its interaction with the discriminator. The discriminator has access to both the synthetic samples and those extracted from the set of real images. The error signal to the discriminator is supplied through the simple ground truth of whether the image originates from the actual set or the generator. The same error signal, through the discriminator, can be used to train the generator, leading it to produce better-quality fakes.

The discriminator is a classifier whose main objective is to obtain the conditional probability $P(y|x)$. It discovers how to model the probability that an example is true or

Figure 9 – The two models which are learned during the training process for a GAN are the discriminator ($\mathcal{D}$) and the generator ($\mathcal{G}$).


Synthetic data sample
x'
Generator is trained to map a noise sample to a synthetic data sample that can beat the discriminator
$\mathcal{G}$
OR
$\mathcal{D}$
Real or fake image?
Discriminator is trained to distinguish real data samples from synthetized samples
z
Noise source
x
Real data sample

Source: The author (2022)

false given that set of input features. The output probabilities of the discriminator are classification labels. The goal of discriminative models is to detect false generated data, so the discriminative neural network is trained to minimize the final classification error. It learns to highlight the different classes by looking at real and fake samples created by the generator and tries to tell which ones are real and which ones are fake (GOODFELLOW et al., 2014).

Networks symbolizing the generator and discriminator are generally applied across multilayer networks consisting of convolutional and/or fully connected layers. The generating and discriminating networks must be distinguishable, even if they need not be directly invertible. Suppose one determines the generating network as a mapping from some representation space, called the latent space, to the data space (images will be the primary objective). In that case, one can denote this more formally as $\mathcal{G} : \mathcal{G}(z) \to \mathcal{R}^{|x|}$, where $z \in \mathcal{R}^{|z|}$ is a sample from the latent space, $x \in \mathcal{R}^{|x|}$ is an image and $| \cdot |$ denotes the number of dimensions (CRESWELL et al., 2018).

3.3 GENERATIVE ADVERSARIAL NETWORKS FOR SUPER RESOLUTION

In this topic, some examples of how GAN models were applied for SR tasks will be described. SRGAN applies a deep network in combination with an adversary network to produce higher-resolution images. To obtain a better and more efficient SR model, a variety of deep learning methods were applied to a large-scale image dataset to solve the ISR tasks. For example, (LEDIG et al., 2017) features a super-resolution convolutional neural network (SRCNN)-based pixel mapping that simply had three layers for greater learning power than other popular SR machine learning methods. of pictures. Although the SRCNN had an excellent SR effect, it still had issues with shallow architecture and high complexity. SRGAN uses the perpetual loss function, which is the weighted sum of two loss components: content loss and adversary loss. This loss is very important for the performance of the generator architecture.

In designing novel generators and discriminators, progressively growing generative adversarial networks, such as PGGANs or ProGANs, used convolutional layers to gradually deepen LR images to improve image qualities for image recognition. An ESRGAN uses residual dense blocks in a generator without batch normalization to extract more detailed information for ISR. To suppress the effects of checkerboard artifacts and unpleasant high frequency, multi-discriminators were proposed for the ISR (LEE; SHIN; KIM, 2019). This means that a perspective discriminator was implemented to overcome the grid artifacts, and a gradient perspective was used to address the high-frequency unpleasant questions in the SR image (WANG et al., 2018c). This architecture is the basis for many other SOTA proposals in SR. Below are three structures that will be the basis for the work of this thesis.

3.3.1 Real-world Super-resolution via Kernel Estimation and Noise Injection

This project proposes an innovative, realistic degradation framework for SR, including kernel estimation and noise injection to preserve the original domain attributes. On the one hand, the authors first use the existing kernel estimation method to generate more realistic LR images. On the other hand, they propose a simple and effective method to directly collect noise from the original image and add it to the reduced image. In addition, the patch discriminator for RealSR is introduced to avoid generated artifacts (JI et al., 2020b).

The RealSR model is trained on paired data constructed $I^{LR}, I^{HR}, \in X, Y$. Moreover, the generator also acquires an RRDB structure based on ESRGAN, an improvement of the SRGAN network structure by introducing the Residual-in-Residual Dense Block RRDB, which is of higher capacity and easier to train (WANG et al., 2018c). The resolution of the generated image is enlarged x4 times. Various losses are applied to training, including pixel loss, perceptual loss, and adversarial loss (JOHNSON; ALAHI; FEI-FEI, 2016).

However, the authors noted that the ESRGAN discriminator could introduce many artifacts, unlike the default setting of ESRGAN. This is why they use a patch discriminator instead of VVG-128 (SIMONYAN; ZISSERMAN, 2014) due to two conveniences: Firstly, VVG-128 limits the size of the generated image at 128, doing multiscale training is inconvenient, and later, VVG-128 contains a deeper network and its fully connected fixed layers make the discriminator pay more attention to global features and ignore local features. This proposal achieved first place in the NTIRE 2020 SR competition (LUGMAYR; DANELLJAN; TIMOFTE, 2020).

3.3.2 Real-ESRGAN: Training Real-world Blind Super-resolution with Pure Synthetic Data

The Real-ESRGAN aims to improve the robust ESRGAN architecture to restore real-world general LR images by synthesizing training pairs with a more pragmatic degradation procedure. Real complex degradations usually come from complicated combinations of different processes and types of degradation, such as camera imaging systems, image editing, and internet streaming. For example, a photo taken with a cell phone may have different degradations, such as camera blur, sensor noise, sharpness artifacts, and Joint Photographic Experts Group (JPEG) compression. In addition, they are edited and uploaded to a social media application, which adds higher compression and unpredictable noise. Therefore, the above process becomes more complicated when the image is shared multiple times on the Internet (WANG et al., 2021b).

This prompts the purpose of this project to extend the classical first-order degradation model to a higher-order type of degradation model for real-world applied degradations, i.e. the degradations are modeled with various other processes. replicated degradation processes, each process being the classical degradation. model. Through a series of experiments, the authors adopt a second-order degradation process for a good balance between simplicity and efficiency. (ZHANG et al., 2021) also proposes a random shuffling strategy to synthesize more practical impairments. However, it still involves a specific number of degradation processes, and it is not sure whether all the shuffled degradations are useful or not. In contrast, higher-order degradation modeling is more flexible and attempts to match the actual degradation generation process. Additionally, it includes sync filters in the synthesis procedure to mimic familiar ringing and overshoot artifacts.

Because the degradation space is much larger than ESRGAN, the training also becomes demanding. Specifically, the discriminator requires a higher force capability to discriminate the reality of complex training outcomes, while the discriminator's gradient feedback needs to be more accurate to improve local detail. Therefore, the authors' research considerably improves the VVG-style discriminator in ESRGAN to a U-Net design. After this, the U-Net structure and complicated degradations also increase the instability of the training. Consequently, spectral normalization (SN) regularization, a widely-used technique for improving the stability and sample quality of Generative Adversarial Networks (GANs), is used to stabilize the training dynamics (SCHONFELD; SCHIELE; KHOREVA, 2020). Equipped with the dedicated enhancements, they finally trained the Real-ESRGAN and achieved a good balance of local detail enhancement and artifact suppression.

3.3.3 A-ESRGAN: Training Real-world Blind Super-resolution With Attention U-Net Discriminators

In this frame of reference, the authors of the architecture A-ESRGAN establish a new discriminator network structure, a multiscale care U-Net discriminator, and add it with the existing RRDB-based generator to build their deep learning network. (WEI et al., 2021b). This model maintains that superior results are obtained over the last generation Real-ESRGAN model in sharpness and detail. This result is due to the conjunction of the attention mechanism. U-Net Structure in its proposed discriminator, similar to the previous proposal Real-ESRGAN, The U-Net structure in the discriminator can provide feedback per pixel to the generator (SCHONFELD; SCHIELE; KHOREVA, 2020), which can help the generator produce higher relief and detail features, such as texture or brush strokes. At the same time, the spotlight layer can not only distinguish the outline of the image area to preserve overall coherence but also strengthen the lines and edges of the image to prevent blurring. Therefore, the combination of U-Net and Attention is a very encouraging scheme. Furthermore, to increase the perceptual field of this discriminator, two attention U-Net discriminators have an identical network structure. Still, they operate at different image scales than the final discriminator, which is called a multiscale discriminator. Extensive experiments show that the present model outperforms most existing GAN models in both quantitative NIQE (MITTAL; SOUNDARARAJAN; BOVIK, 2012) performance metrics and qualitative image perception sensations.

4 ARCHITECTURE

4.1 INTRODUCTION

In this work, the main pipeline is divided into three parts. First of all, inspired by the work of RealSR architecture with a novel degradation framework design for real-world images by estimating various blur kernels, as well as real noise distributions, it was applied a realistic degradation process to extract the noise from real-world images so they are injected into the process of generating LR images to make them more realistic as they are presented into RWSR tasks (JI et al., 2020b). Secondly, the following stage of this project is to implement the SR model based on GAN architecture with the previously constructed data. Therefore, the base of this project will be the previously mentioned GAN architecture which is ESRGAN. Furthermore, several experiments were performed adding a new LPIPS perceptual loss (ZHANG et al., 2018) to the original arrangement of loss functions in the generator module because this loss function reflects more appropriately the human perception preferences than the VGG perceptual loss and encourages natural and perceptually pleasing results (JO; YANG; KIM, 2020). Finally, and the most remarkable section of this work will be to perform a modification in the main GAN architecture, adding a U-Net structure discriminator (SCHONFELD; SCHIELE; KHOREVA, 2020) to consider both the global and local context of an input image including a CutMix process for data augmentation during training.

4.2 DATA PRE-PROCESSING

First, the source domain X will be defined as the original real images and the clean HR images as the target domain Y. Blur kernels with different degrees directly affect the blur of the downsampled images. Bicubic can be considered an ideal way of downsampling because it retains the information from X as much as possible. Nonetheless, the frequency of these downsampled images has changed to another domain X. When training on $\{X, Y\}$, the model will try to recover all the details due to all information being essential in the domain X. Usually, a SR model works well on LR images but usually fails on $I_{src} \in X$, which is an unprocessed real image. Another problem is the downsampled image has almost no noise, while real-world images in X usually have a lot. Mere estimation of the blurry kernel cannot accurately model the degradation process. Therefore, in the pre-processing stage, UR-SRGAN and the other SOTA SR models models will be using three different datasets for training: DF2K, DIV2K, and DF2K+OST. These datasets will be referenced as the target Y because it counts on a complete set of LR images, each of them with their respective HR image. In addition, the DPED dataset is introduced (IGNATOV et al., 2017), a large-scale dataset that consists of photos taken synchronously in the wild by three smartphones and one DSLR camera. This set will be considered the source dataset X since different camera devices took all these images in real-world situations. These image datasets will be used to work under kernel estimation and noise injection to create a new set of LR and HR training images and will be better explained in the next section.

4.2.1 Realistic Degradation for Super-resolution

This part will focus on explaining the implementation of a proficient real image degradation based on kernel estimation and noise injection. As mentioned in the previous chapter, the LR image is obtained by the following degradation method 2.4. To estimate the degradation method more precisely, the kernel and noise are estimated from the image. After getting the estimated kernel and noise patch, a degradation pool is created, which is used to degrade clean HR images into blurry and noisy images, thus generating image pairs for training SR models. To achieve this process, first of all, a kernel and noise estimation will be initiated onto the real-world images dataset specified as the source X to obtain a pool collection from these two elements from the images. Subsequently, a bicubic interpolation will be performed in the HR images provided by the target Y dataset to retrieve LR clean images. Once both LR and HR is created from the target, the pre-processing phase is complete to start the training stage. During this phase, the kernels and noises collected previously will be added to the LR generated images to simulate real-world images. To describe the realistic degradation method concisely, it is better described as an algorithm shown in Algorithm 1.

Algorithm 1 Data-constructing pipeline

```
Require: Real images set $X$ , $HR$ images set $Y$, downsampling scale factor $s$
Ensure: Realistic paired images $\{I_{LR}, I_{HR}\}$
Initialize kernel pool $K = 0$
Initialize noise pool $N = 0$
for all $I_{src}$ such that $I_{src} \in X$ do
    Estimate $k$ from $I_{src}$ by solving Equation: 4.1
    Add $k$ to $K$
    Crop $n$ from $I_{src}$
    if $n$ meets Equation: 4.4 then
        Add $n$ to $N$
    end if
end for
for all $I_{HR}$ such that $I_{HR} \in Y$ do
    Randomly select $k_i \in K, n_j \in N$
    Generate $I_{LR}$ with $k_i$ and $n_j$
end for
```

4.2.2 Kernel Estimation and Downsampling

To perform the task of estimating the adequate kernel from a real image (taking into consideration that this use case is a BSR problem), it is required a kernel estimation algorithm to estimate kernels from real images explicitly. That is why the KernelGAN proposal was adopted to use its performance for the kernel estimation method and set appropriate parameters based on real images. Its generator is trained to produce a downscaled version of the LR test image, such that its discriminator is not capable of distinguishing between the patch distribution of the downscaled image and the patch distribution of the original LR image (BELL-KLIGLER; SHOCHER; IRANI, 2019). The estimated kernel needs to meet the following constraints shown in Equation 4.1:

$$\text{argmin}_k \parallel (I_{src} * k) \downarrow_s - I_{src} \downarrow_s \|_1 + \mid 1 - \Sigma k_{i,j} \mid + \mid \Sigma k_{i,j} \cdot m_{i,j} \mid + \mid 1 - D((I_{src} * k) \downarrow_s) \mid, \tag{4.1}$$

Where $(I_{src} * k) \downarrow_s$ is downsampled LR image with kernel k, and $I_{src} \downarrow_s$ is downsampled image with the ideal kernel, therefore to minimize this error is to encourage the downsampled image to preserve important low-frequency information of the source image. Furthermore, the second term of the above formula is to constrain k to sum to 1, and the third term is to penalty boundaries of k. Eventually, the discriminator $D(\cdot)$, in this case, implemented as a U-Net discriminator, is to ensure the consistency of the source domain.

4.2.3 Cleaning-up

To get more HR images, it is required to generate noise-free images from X. Specifically, it is adopted a bicubic downsampling strategy on the real image in the source domain to remove noise and make the image sharper. Let $I_{src} \in X$ be an image from real source images set, and k_{bic} be the ideal bicubic kernel. Then the image is downsampled with a clean-up scale factor sc. This can be seen in Equation 4.2

$$I_{HR} = (I_{src} * k_{bic}) \downarrow_{sc} . \tag{4.2}$$

4.2.4 Degradation with Blur Kernels

Images after resolution reduction are processed as clean HR images. These HR images are then degraded by randomly selecting a defocus core from the degraded pool. The downsampling process consists of cross-correlation operations followed by stride sampling, which can be formulated as shown in Equation 4.3:

$$I_D = (I_{HR} * k_i) \downarrow_s, i \in \{1, 2...m\}, \tag{4.3}$$

where I_D denotes the downsampled image, and k_i refers to the selected specific blur kernel from $\{k_1, k_2...k_m\}$.

4.2.5 Noise Injection

For noisy images, it will be explicitly injected noise into the downsampled images to generate realistic LR images. Since the high-frequency information is lost during the downsampling process, the degraded noise distribution is modified at the same time. Thus, to make the degraded image have a similar noise distribution to the source image, there is a process to collect noise patches from the source dataset X. Here, it is seen that patches with strong content have a more considerable variance. It was implemented a filtering rule to collect patches with their variance in a particular range inspired by the work of (CHEN et al., 2018); a novel two-step framework with a Generative Adversarial Network GAN to estimate the noise distribution over the input noisy images and the noise patches to construct a paired training dataset, and (ZHOU; SUSSTRUNK, 2019); a kernel modeling super-resolution network (KMSR) that incorporates a pool of realistic blur-kernels with a generative adversarial network GAN to train a SR network with HR and corresponding LR images constructed with the generated kernels. Consequently, noise and content from the images are detached by the following rule shown in Equation 4.4:

$$\sigma(ni) < \upsilon, \tag{4.4}$$

where $\sigma(\cdot)$ denotes the function to calculate variance, and υ is the max value of variance.

4.2.6 Degradation with Noise Injection

Assuming that a series of noise patches $n_1, n_2...n_l$ are collected and added to the degradation pool. The noise injection process is performed by randomly cropping patches from the noise pool. This process is expressed as in Equation 4.5:

$$I_{LR} = I_D + n_i, i \in \{1, 2...l\}, \tag{4.5}$$

where n_i is a cropped noise patch from the noise pool consisting of $k_1, k_2...k_l$. In detail, it is assumed a connected noise injection method that the content and the noise are combined during the training phase. This makes the noise more diverse and regularizes the SR model to distinguish content with noise. After the degradation with blur kernels and injecting noise, Thus, it is obtained $I_{LR} \in X$.

4.3 ARCHITECTURE

4.3.1 Generator

As previously mentioned, the architecture is based on a GAN architecture that counts with the generator. The new proposal that will be further expanded in this part is the U-Net Discriminator. For the generator network, it is used the same structure used on ESRGAN proposal for this SR model and later on, trained on constructed paired data $\{I_{LR}, I_{HR}\} \in \{X, Y\}$. The generator has an RRDB structure, and the resolution of the output image by the generator will be four times larger. ESRGAN removes batch normalization layers from SRGAN to avoid unpleasant artifacts and replaces the original residual block with the RRDB to boost performance (WANG et al., 2018b). Formally, the generator will return ×4 super-resolved output image I^{Gen} from an input image I^{I_n}, as shown in Equation 4.6:

$$I^{Gen} = G(I^{I_n}) \tag{4.6}$$

One part that is very relevant in this proposal about the generator is the employment of multiple losses that are applied to the training process. Concisely, the losses implemented for the generator, usually by other SRGAN methods, are pixel loss, perceptual loss, and adversarial loss. The pixel loss L_1 uses L_1 distance. Perceptual loss L_{per} uses the inactive features of VVG-19, which will help to improve the visual effect of low-frequency features such as edges (SIMONYAN; ZISSERMAN, 2014). Adversarial loss L_{adv} has the task of enhancing the texture details of the generated image to make it look more realistic. However, this adversarial function will be adapted to implementing the U-Net discriminator and will be explained in the next section. A λ value is added to each loss function due to the coefficient controlling how much the regularization term contributes to the total loss function. Therefore, the total loss function would be the weighted sum of all the previous losses, shown in Equation 4.7:

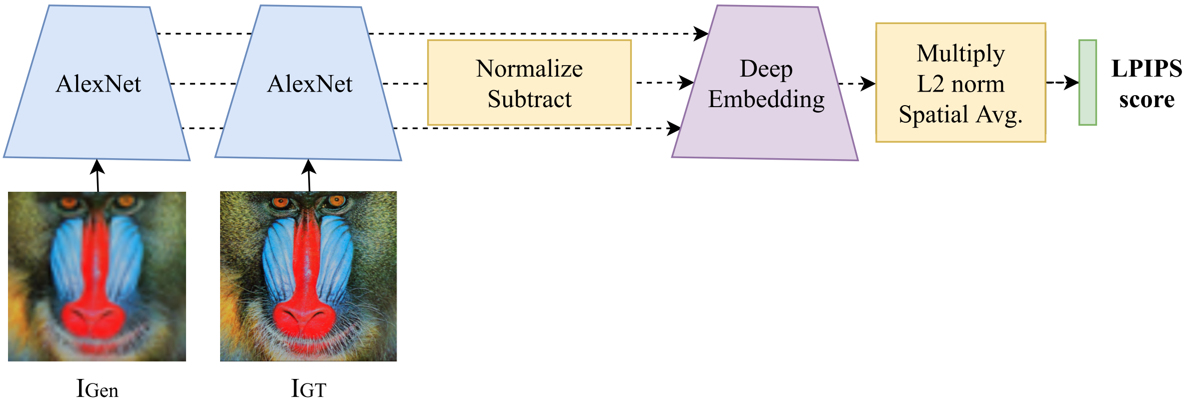
$$L_{total} = \lambda_1 \cdot L_1 + \lambda_{per} \cdot L_{per} + \lambda_{adv} \cdot L_{adv} \tag{4.7}$$

Nonetheless, as several related works are mostly based on the GAN with the VVG perceptual loss, there were few considerations about the loss functions. Therefore, a new strategy is to experiment with the LPIPS loss functions for perceptual extreme SR and instead of replacing the VVG perceptual loss with the LPIPS perceptual loss, Thus, it was added to 4.7 to see if the enhancement of LPIPS is considerable respect to other real-world metric evaluations (LUGMAYR; DANELLJAN; TIMOFTE, 2020). To this end, LPIPS is used for the perceptual loss, shown in Equation 4.8:

$$L_{lpips} = \sum_{k} \tau^k(\phi^k(I^{Gen}) - \phi^k(I^{GT})), \tag{4.8}$$

where ϕ is a feature extractor, τ transforms deep embedding to scalar LPIPS score, and the score is computed and averaged from k layers. To explain further, there is a reference image, then the image is transformed in two different ways small translation and blurring. Traditional image quality metrics like PSNR and SSIM prefer blurred images, but humans are more likely to prefer translated ones (ZHANG et al., 2018). LPIPS is trained with a dataset of human perceptual similarity judgments and more appropriately reflects the human perception preferences than the VVG perceptual loss as appreciated in Figure 10.

Figure 10 – LPIPS is computed from deep feature embeddings.



Source: The author (2022)

In addition, it is added the discriminator's feature matching loss L_{fm}, as shown in Equation 4.9, to alleviate the undesirable noise from the adversarial loss where D^l denotes the activations from the $l-th$ layer of the discriminator D, and H is the Huber loss, or smooth L_1 loss, that is a loss function used in robust regression, that is less sensitive to outliers in data than the squared error loss. (JO; YANG; KIM, 2020):

$$L_{fm} = \sum_{l} H(D^l(I^{Gen}), D^l(I^{GT})), \tag{4.9}$$

As a result, the final generator loss function is shown in Equation 4.10:

$$\begin{multline} L_{total} = \lambda_1 \cdot L_1 + \lambda_{per} \cdot L_{per} + \lambda_{adv} \cdot L_{adv} \\ + \lambda_{fm} \cdot L_{fm} + \lambda_{lpips} \cdot L_{lpips} \end{multline} \tag{4.10}$$

4.3.2 Discriminator

Broadly, many GAN-based SR methods have implemented an encoder structure as discriminators. Since it is simply a classifier, it tries to distinguish real data from the data created by the generator. Thus, this component might use any other network architecture appropriate to the type of data it's classifying. For instance, in RealSR (JI et al., 2020a), they use a Patch Discriminator, which focuses on each N×N patch of the image, and determines if it is real or fake, only penalizing the structure at the scale of local image patches (ISOLA et al., 2017), instead of VVG-128 (SIMONYAN; ZISSERMAN, 2014) because it limits the size of the generated image to 128. Also, it makes the discriminator pay more attention to global features and ignore local features. Even if this approach has optimal performance, the main objective of this project is to implement a new scheme with a U-Net structure discriminator because, according to recent research in this area, it would achieve better perceptual feature extraction considering both the global and local context and giving effective feedback to the generator during the training.

4.3.2.1 U-Net GAN Model

A usual GAN architecture consists of two networks: a generator G and a discriminator D, trained by minimizing the following competing objectives in an alternating manner, as shown in Equations 4.11 and 4.12:

$$\mathcal{L}_D = -\mathbb{E}_x[\log D(x)] - \mathbb{E}[\log(1 - D(G(z)))], \tag{4.11}$$

$$\mathcal{L}_G = -\mathbb{E}_z[\log D(G(z))], \tag{4.12}$$

G proposes to map a latent variable $z \sim p(z)$ sampled from a prior distribution to a realistic-looking image, whereas D aims to distinguish between real x and generated $G(z)$ images. Generally, G and D are modeled as a decoder and an encoder convolutional network, respectively.

Although there are many variations of the GAN objective function and network architectures that were implemented in SR models, the main objective of this document is to improve the discriminator network. Based on the approach of (SCHONFELD; SCHIELE; KHOREVA, 2020), this module will explain the details of the new modification proposed to replace architecture D from a standard classification network to a U-Net encoder-decoder network in a SR oriented GAN model. This modification will be made only in the discriminator module, leaving intact the underlying basic architecture, the generating part that is the encoder. The proposed discriminator allows for maintaining the representation of global and local data, providing more informative feedback to the generator.

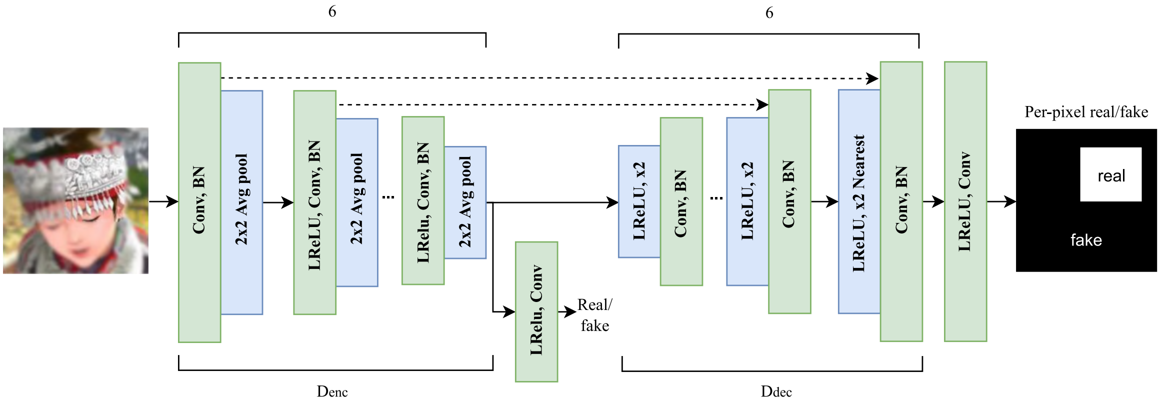
With the advantage of the local feedback per pixel of the new proposed U-Net decoder module, a consistency regularization technique will be applied, penalizing the inconsistent predictions per pixel of the discriminator by means of the CutMix transformations of images (real and fake). This technique helps to improve the localization quality of the U-Net discriminator and induces it to pay more attention to semantic and structural changes between real and fake samples. Another advantage of this method, as mentioned above, is that it is compatible with most GAN models since it does not modify the generator in any way and leaves the original GAN target intact (YUN et al., 2019).

4.3.2.2 U-Net-based Discriminator

Encoder-decoder networks constitute an effective method for dense prediction. U-Nets, in particular, have shown very high performance on many complex image segmentation tasks. In these methods, similar to image classification networks, the encoder progressively downsamples the input, capturing the context of the big picture. The decoder progressively oversamples, matching the output resolution to the input resolution, thereby achieving precise localization. Skip connection routes data between the matching resolutions of the two modules, further enhancing the network's ability to segment fine detail accurately. This is how it is proposed to extend a discriminator to form a U-Net, reusing the essential components of the original discriminator classification network as part of the encoder and the components of the generator network as part of the decoder (RONNEBERGER; FISCHER; BROX, 2015).

On top of the standard encoder structure D_{enc}, they successively attached a decoder structure D_{dec} for providing per-pixel feedback to the generator. while maintaining global context as shown in Figure 11. In other words, the discriminator now consists of the original downsampling network and a new upsampling network. The two modules are connected through bottleneck and jump connections that copy and concatenate the encoder and decoder feature map modules.

Figure 11 – To provide per-pixel feedback to the generator, U-Net discriminator structure is adopted. There are 6 downsampling and 6 upsampling stages, with skip-connections between them.

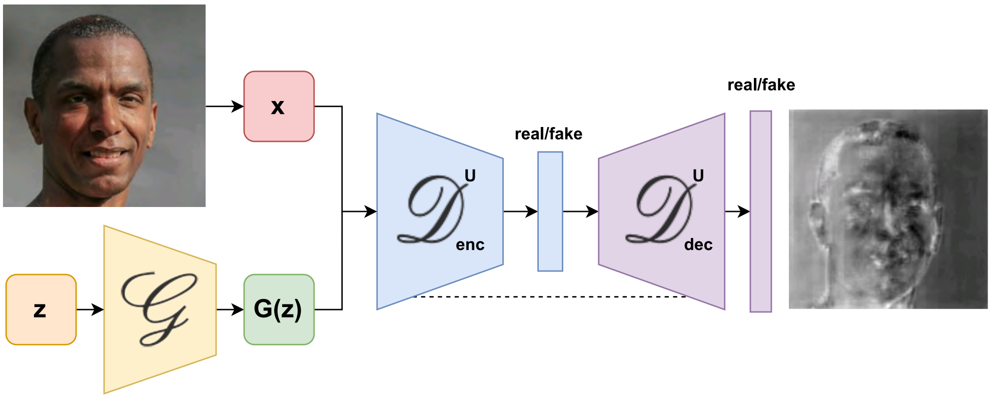


Source: The author (2022)

While the original $D(x)$ classifies the input image x into being real and fake, the U-Net discriminator $D^U(x)$ additionally performs this classification on a per-pixel basis, segmenting image x into real and fake regions, along with the original image classification of x from the encoder, Please, refer to Figure 13 to observe this more precisely. This enables the discriminator to learn both global and local differences between real and fake images.

The U-Net discriminator encoder network acts as the feature extractor and learns an abstract representation of the input image through a sequence of the encoder blocks. Each encoder block consists of an extraction of [64, 128, 192, 256, 320, 384] feature channels applying skip connections and the input is a 3-channel image. Consequently, the U-Net discriminator decoder network is used to take the abstract representation and generate a semantic segmentation mask. The decoder block consists of [384, 320, 256, 192, 128, 64] feature channels. To review the code implementation of the U-Net Discriminator, please refer to A.1 for more detailed information.

Figure 12 – U-Net GAN. The proposed U-Net discriminator classifies the input images on a global and local per-pixel level. Due to the skip connections between the encoder and the decoder (dashed line), the channels in the output layer contain both high- and low-level information. Brighter colors in the decoder output correspond to the discriminator confidence of the pixel being real (and darker of being fake).



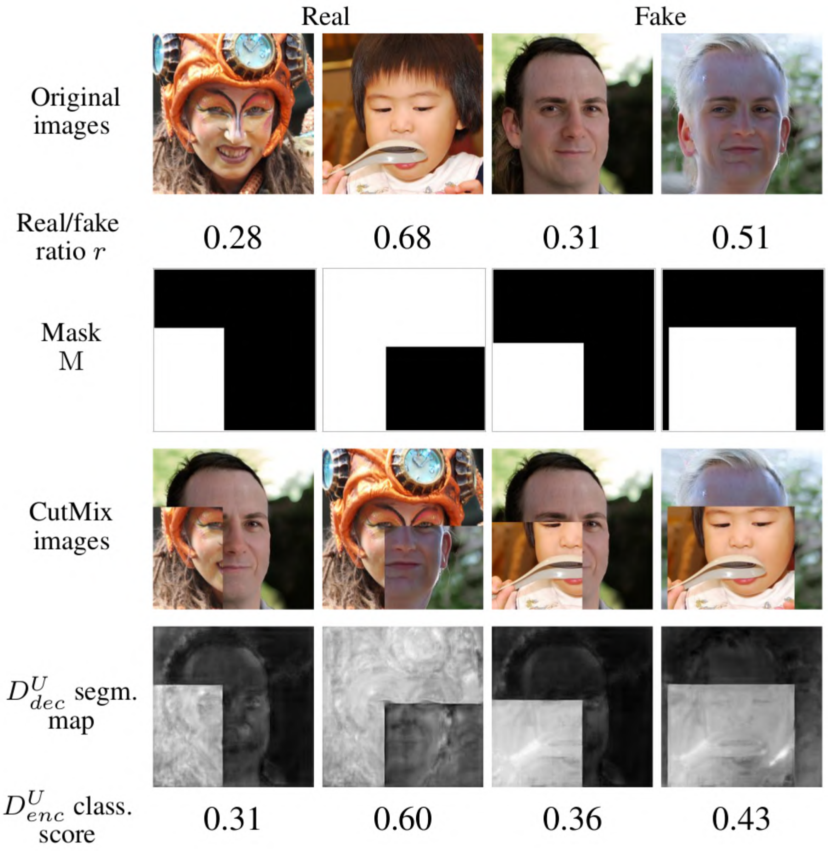
Source: The author (2022)

4.3.2.3 Consistency Regularization

In this section, the consistency regularization technique for the U-Net-based discriminator presented in the previous section will be further developed. The per-pixel decision of the correctly trained D^U discriminator has to be equivalent under any kind of image transformation that alters the class domain. However, this characteristic is not expressly guaranteed. To be enabled, the discriminator must be regularized to focus more on semantic and structural changes between the real and false samples and pay less attention to arbitrary perturbations that preserve class dominance.

Consequently, applying the consistency regularization technique of the D^U discriminator is proposed, encouraging the D^U decoder module to generate equivalent predictions under the CutMix transformations (YUN et al., 2019) of real and false samples. CutMix augmentation generates synthetic images by cutting and pasting image patches of different classes. This strategy is the most optimal choice for this work, rather than employing previous alternatives such as MixUp or CutOut, which focus on penalizing the classification network sensitivity to samples generated images. (ZHANG et al., 2017), because it does not alter the real and fake image patches used to perform the mix, preserving its original class domain, and it provides a wide variety of possible outputs. The CutMix augmentation strategy and D^U predictions are described better in Figure 13.

Figure 13 – Visualization of the CutMix augmentation and the predictions of the U-Net discriminator on CutMix images. 1st row: real and fake samples. 2nd and 3rd rows: sampled real/fake CutMix ratio r and corresponding binary masks M (color code: white for real, black for fake). 4th row: generated CutMix images from real and fake samples. 5th and 6th row: the corresponding real/fake segmentation maps of D^U with its predicted classification scores.



Source: (YUN et al., 2019)

Following, it is synthesized a new training sample $\tilde{x}$ for the discriminator D^U by mixing x and $G(z) \in \mathbb{R}^{W \times H \times C}$ with the mask M, as shown in Equation 4.13:

$$\begin{aligned} \tilde{x} &= mix(x, G(z), M), \\ mix(x, G(z), M) &= M \odot x + (1 - M) \odot G(z), \end{aligned} \tag{4.13}$$

where $M \in \{0, 1\}^{W \times H}$ is the binary mask that indicates if the pixel (i, j) comes from the real image ($M_{i,j} = 1$) or false ($M_{i,j} = 0$), 1 is a binary mask filled with ones and is multiplication by elements. The class label $c \in \{0, 1\}$ for the new CutMix image $\tilde{x}$ is set to false, i.e. $c = 0$. Globally, the mixed synthetic image should be recognized as false by the D^U_{enc} encoder; otherwise, the generator can learn to introduce CutMix augmentation into the generated samples, causing unwanted artifacts. The synthetic samples $\tilde{x}, c = 0$, and M are the ground truth for the encoder and decoder modules of the D^U discriminator, respectively.

Using the CutMix operation in 4.13, the discriminator is trained to provide consistent per-pixel predictions, i.e. $D^U_{dec}(mix(x, G(z), M)) \approx mix(D^U_{dec}(x), D^U_{dec}(G(z)), M)$, by establishing the consistency regularization loss term in the discriminator purpose, shown in Equation 4.14:

$$\mathcal{L}^{cons}_{D^U_{dec}} = \| D^U_{dec}\Big(mix(x, G(z), M)\Big) - \Big(mix(D^U_{dec}(x), D^U_{dec}(G(z)), M)\Big) \|^2 \tag{4.14}$$

where the notation for the L^2 norm is $\| \cdot \|$. This consistency loss is obtained between the per-pixel output of D^U_{dec} on the CutMix image and the CutMix between outputs of the D^U_{dec} on real and fake images, penalizing the discriminator for inaccurate prediction.

4.3.2.4 U-Net Discriminator Loss Functions

About the U-Net discriminator loss functions, Above the normal encoder structure D_{enc} there is a decoder structure D_{dec} for providing per-pixel feedback to the generator while preserving global context. The discriminator loss L_D is computed at both the encoder head $L_{D_{enc}}$ and the decoder head $L_{D_{dec}}$. The formulation for the discriminator loss as hinge loss that was applied is shown in Equations 4.15 and 4.16:

$$\begin{aligned} L_{D_{enc}} = -\mathbb{E}\left[\sum_{i,j} min\left(0, -1 + \Big[D_{enc}(I^{GT})\Big]_{i,j}\right)\right] \\ - \mathbb{E}\left[\sum_{i,j} min\left(0, -1 - \Big[D_{enc}(I^{Gen})\Big]_{i,j}\right)\right] \end{aligned} \tag{4.15}$$

$$\begin{aligned} L_{D_{dec}} = -\mathbb{E}\left[\sum_{i,j} min\left(0, -1 + \Big[D_{dec}(I^{GT})\Big]_{i,j}\right)\right] \\ - \mathbb{E}\left[\sum_{i,j} min\left(0, -1 - \Big[D_{dec}(I^{Gen})\Big]_{i,j}\right)\right] \end{aligned} \tag{4.16}$$

where I^{GT} is the ground truth image, and $[D(I)]_{i,j}$ is the discriminator decision at pixel (i, j). Besides, the adversarial loss for the generator is shown in Equation 4.17:

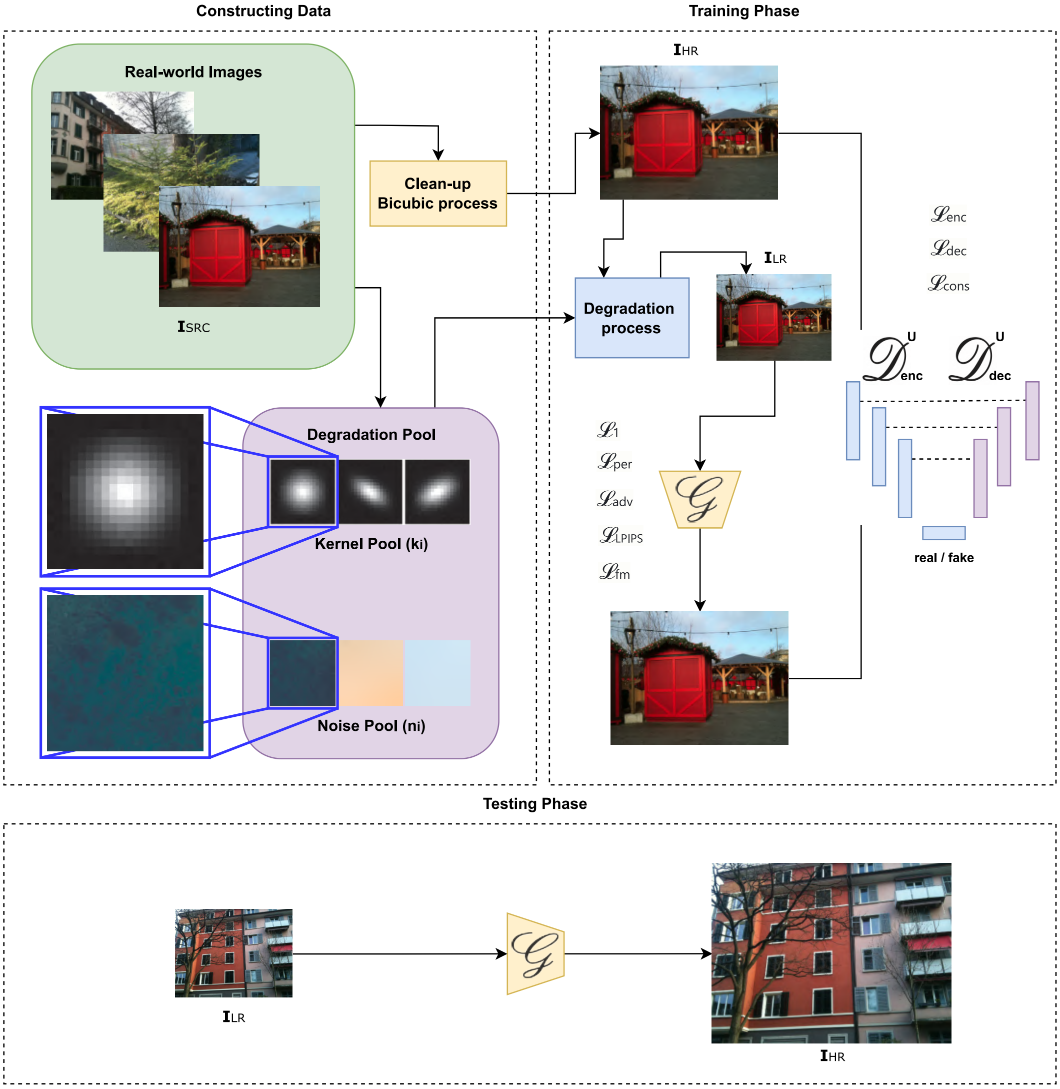
$$L_{adv} = -\mathbb{E}\left[\sum_{i,j} \Big[D_{enc}(I^{GT})\Big]_{i,j} + \sum_{i,j} \Big[D_{dec}(I^{Gen})\Big]_{i,j}\right] \tag{4.17}$$

Additionally, the consistency regularization loss function was applied to the discriminator in order to synthesize the training samples by using CutMix transformation and minimizing the loss $L_{D_{cons}}$ (YUN et al., 2019). Finally, the total discriminator loss is shown in Equation 4.18:

$$L_D = L_{D_{enc}} + L_{D_{dec}} + L_{D_{cons}} \tag{4.18}$$

In Figure 14, the global architecture of UR-SRGAN is shown, identifying the training and testing phase. In addition, having a better perception of the pre-processing by generating LR images, the loss functions, and the implementation of the U-Net block as a discriminator.

Figure 14 – The framework of the UR-SRGAN architecture. The degradation pool provides diverse blur kernels and noise distributions for constructing realistic LR images. During the training phase, the SR model is optimized to reconstruct HR images.



Source: The author (2022)

5 OVERALL EXPERIMENTS AND HYPOTHESIS VERIFICATION

5.1 INTRODUCTION

To test the proposed hypothesis of adding additional loss functions in the generator and implementing a U-Net architecture as a discriminator, a series of experiments were carried out to verify the effectiveness of this proposal with respect to other proposals that are in the SOTA when performing tasks of SR in single-images.

For this, three primary databases were used (DIV2K, DF2K, and DF2K+OST) that will be described in more detail in this chapter. In addition, the characteristics of the computer in which the experiments were executed and the technologies, libraries, and versions that were required for the model to be trained will be described. For training, the hyperparameters, the number of epochs required, and the number of parameters used by the UR-SRGAN model and the other three models are taken into account to compare the results obtained by the proposal of this work.

This chapter will be divided into four sections. In the first section, all the details of how the experiments of the architecture proposed in this work (UR-SRGAN) were developed will be specified, and it will be shown if this managed to achieve the initial objectives that, are to obtain better metrics than other solutions in the SOTA for SR of single-images. Subsequently, the other three sections describe how the other three reference models were trained with the same image databases but with the parameters, hyperparameters, and configuration that they originally have in order to have a fair comparison between all the proposals.

5.2 DATASETS

For the training phase of the four models to be executed, that is the proposal of this project UR-SRGAN and the other references (RealSR, Real-ESRGAN, and A-ESRGAN) will be trained using the three different image databases and for the test phase, the test data from DIV2K dataset, consisting of 100 LR images are available, each of them with a ground-truth to evaluate the results. These datasets are detailed below:

- DIV2K [1]: A large dataset of RGB images with a large diversity of contents. The DIV2K dataset is divided into:
  - Train data: starting from 800 high definition HR images we obtain corresponding LR images and provide both HR and LR images for 2, 3, and 4 downscaling factors. All experiments will explicitly use only 4 scale factors.

[1] Available at <https://data.vision.ee.ethz.ch/cvl/DIV2K/>, 2022.

  - Validation data: 100 high definition HR images are used for generating LR corresponding images.
  - Test data: 100 diverse images are used to generate LR corresponding images.

- DF2K [2]: The DF2K dataset merges the DIV2K (TIMOFTE et al., 2017) and Flickr2K (AGUSTSSON; TIMOFTE, 2017) datasets, and contains a total of 3450 images. These images are artificially added with Gaussian noise to simulate sensor noise. The validation set contains the same 100 images from DIV2K with corresponding ground truth, therefore the metrics based on reference can be calculated.

- DF2K+OST [3]: A Kaggle dataset that contains the previous DF2K dataset with an extra of about 9000 smaller obtained from the OST dataset. Having a total of 12434 images. The validation set contains the same 100 images from DIV2K with corresponding ground truth, therefore the metrics based on reference can be calculated.

- DPED [4]: The DPED (IGNATOV et al., 2017) dataset contains 5614 images taken by the iPhone3 camera. The images in this dataset are unprocessed real images, which are more challenging containing noise, blur, dark light, and other low-quality problems. The 100 images in the validation set are cropped from original real images. Since there is no corresponding ground truth, it will only be provided an evaluation based on the NIQE (MITTAL; SOUNDARARAJAN; BOVIK, 2012) metrics and visual comparison.

5.3 EXPERIMENTS FOR UR-SRGAN

5.3.1 Training Details

This architecture was implemented in PyTorch 1.7.1 and trained on a single NVIDIA GeForce RTX 2080 Ti (12G). Furthermore, the generator was trained for about 60K iterations with a mini-batch size of 16. As (JI et al., 2020b) was a main reference for this implementation, the values for lambda values: $\lambda_1 = 1E^{-2}$, $\lambda_{per} = 1$, $\lambda_{adv} = 1E^{-3}$, $\lambda_{lpips} = 1E^{-3}$, and $\lambda_{fm} = 1$ were applied empirically. Adam optimizer was selected for this work and the learning rate is set to 0.0001 for training both the generator and the discriminator networks. Three training experiments were developed on the referred datasets (DIV2K, DF2K, DF2K + OST).

This proposal was to add new loss functions for the generator and replace the discriminator with a U-Net discriminator. That is why it was performed in three main phases during the UR-SRGAN workflow. Initially, the model was trained only with the appended

[2] Available at <https://competitions.codalab.org/competitions/22220>, 2022.
[3] Available at <https://www.kaggle.com/datasets/thaihoa1476050/df2k-ost>, 2022.
[4] Available at <https://people.ee.ethz.ch/ ihnatova/>, 2022.

LPIPS and Feature Matching losses without modifying the base discriminator that ESRGAN presents. Later on, the U-Net discriminator is added and run in another training phase. Lastly, the last training phase is executed with the combination of both the new losses for the generator and the U-Net discriminator. Only during this last experiment, it could achieve important improvement with respect to the other metrics obtained by the other reference models. Table 2 shows the datasets, training time, parameters for generator and discriminator for the training of the UR-SRGAN model.

Table 2 – Training time consumed and parameters quantity for UR-SRGAN model.

Dataset	**Training time (hh:mm:ss)**	**Parameters G**	**Parameters D**
DIV2K	08:31:36	16,697,987	12,823,810
DF2K	10:03:38	16,697,987	12,823,810
DF2K + OST	08:18:04	16,697,987	12,823,810

Source: The author (2022)

5.3.2 Testing Phase

After training the UR-SRGAN model, the generator and parameters are already trained with the weights ready for the inference process. Thus, it is used to generate 100 new SR images using the test set, with the corresponding ground truth images, from the DIV2K dataset, Consequently, the metrics (PSNR, SSIM, and LPIPS) are calculated. The experiments demonstrated that this new architecture and modifications have a significant improvement from the other three SR methods that were selected to compare its metrics. Especially, reaching better LPIPS performance, denoting our results are much closer to the ground truth relating to visual characteristics. Table 3 shows the datasets, and the metrics for the trained UR-SRGAN model, including the mean score value for each evaluation.

Table 3 – UR-SRGAN Inference Results on the 100 test images in DIV2K dataset. Quantitative results for the UR-SRGAN model compared with the three selected datasets in which training was carried out.↑ and ↓ mean higher or lower is desired.

Dataset	***PSNR* ↑**	***SSIM* ↑**	***LPIPS* ↓**	***Mean Score (sec)***
DIV2K	26.24	0.734	0.228	159.1
DF2K	26.15	0.723	0.221	152.7
DF2K + OST	26.04	0.726	0.225	159.8

Source: The author (2022)

5.4 EXPERIMENTS FOR REAL-SR

5.4.1 Training Details

This proposal was implemented in PyTorch 1.7.1 and trained on a single NVIDIA GeForce RTX 2080 Ti (12G). In the same way as UR-SRGAN, the generator was trained for about 60K iterations with a mini-batch size of 16. Finally, the same hyperparameters selected by (JI et al., 2020b) were used in order to achieve the same results as the authors for the three different datasets in order to get the same results to perform a fairly comparison between all the outcome metrics.

For this phase, the (JI et al., 2020b) the model was trained with the three image databases previously mentioned: (DIV2K, DF2K and DF2K+OST). Furthermore, the same original hyperparameters and settings were used to achieve the same metrics that the authors published training on the original dataset DF2K. The empirical values of lambda selected by the authors for the loss functions were: $\lambda_1 = 1E^{-2}$, $\lambda_{per} = 1$, and $\lambda_{adv} = 1E^{-3}$. Adam optimizer was selected for this work and the learning rate is set to 0.0001 for training both the generator and the discriminator networks. Table 4 shows the datasets, training time, parameters for generator and discriminator for the training of the RealSR model.

Table 4 – Training time consumed and parameters quantity for RealSR model.

Dataset	Training time (hh:mm:ss)	Parameters G	Parameters D
DIV2K	08:10:05	16,697,987	14,499,401
DF2K	09:21:56	16,697,987	14,499,401
DF2K + OST	08:08:41	16,697,987	14,499,401

Source: The author (2022)

5.4.2 Testing Phase

After training the RealSR model, the generator and parameters are already trained with the weights ready for the inference process. Thus, it is used to generate 100 new SR images using the test set, with the corresponding ground truth images, from the DIV2K dataset, Consequently, the metrics (PSNR, SSIM, and LPIPS) are calculated. The new experiments with the new databases gave quite good results, even with different databases. Checking the efficiency of kernel estimation and noise collection of real-world images. Table 5 shows the datasets, and the metrics for the trained RealSR model, including the mean score value for each evaluation.

Table 5 – RealSR Inference Results on the 100 test images in DIV2K dataset. Quantitative results for the RealSR model compared with the three selected datasets in which training was carried out.↑ and ↓ mean higher or lower is desired.

Dataset	***PSNR*** ↑	***SSIM*** ↑	***LPIPS*** ↓	***Mean Score (sec)***
DIV2K	25.08	0.701	0.237	151.1
DF2K	24.83	0.672	0.227	138.7
DF2K + OST	24.63	0.687	0.244	143.0

Source: The author (2022)

5.5 EXPERIMENTS FOR REAL-ESRGAN

5.5.1 Training Details

This architecture was implemented in PyTorch 1.7.1 and trained on a single NVIDIA GeForce RTX 2080 Ti (12G). Similar to ESRGAN, the authors adopt DF2K and OST dataset. In this work, also the three datasets (DIV2K, DF2K, and DF2K+OST) were used for the training phase. This model was trained with the same conditions as the original implementation. The training HR patch size is set to 256. Originally, the total batch size to train this network was 48. However, due to computation limits, it had to be reduced to 8. Also, it was used the Adam optimizer.

Moreover, Real-ESRNet is finetuned from ESRGAN for faster convergence. In addition, Real-ESRNet was trained for 1000K iterations with a learning rate $2E^{-4}$ while training Real-ESRGAN for $400K$ iterations with a learning rate $1E^{-4}$. It adopted an exponential moving average (EMA) for more stable training and better performance. RealESRGAN is trained with a combination of L_1 loss, perceptual loss, and GAN loss, with weights $\{1, 1, 0.1\}$, respectively. Also it has the $\{conv1, ...conv5\}$ feature maps (with weights $\{0.1, 0.1, 1, 1, 1\}$) before activation in the pre-trained VVG-19 network as the perceptual loss. The implementation was also based originally on the BasicSR, an open-source image and video restoration toolbox based on PyTorch, such as super-resolution, denoise, deblurring, and JPEG artifacts removal (WANG et al., 2018a). Table 6 shows the datasets, training time, parameters for generator and discriminator for the training of the Real-ESRGAN model.

5.5.2 Testing Phase

After training the Real-ESRGAN model, the generator and parameters are already trained with the weights ready for the inference process. Thus, it is used to generate 100 new SR images using the test set, with the corresponding ground truth images, from the DIV2K dataset, Consequently, the metrics (PSNR, SSIM, and LPIPS) are calculated. Table 7

Table 6 – Training time consumed and parameters quantity for Real-ESRGAN model.

Dataset	Training time (hh:mm:ss)	Parameters G	Parameters D
DIV2K	1 day, 12:02:58	16,697,987	4,376,897
DF2K	1 day, 10:20:49	16,697,987	4,376,897
DF2K + OST	1 day, 09:13:33	16,697,987	4,376,897

Source: The author (2022)

shows the datasets, and the metrics for the trained Real-ESRGAN model, including the mean score value for each evaluation.

Table 7 – Real-ESRGAN Inference Results on the 100 test images in DIV2K dataset. Quantitative results for the Real-ESRGAN model compared with the three selected datasets in which training was carried out.↑ and ↓ mean higher or lower is desired.

Dataset	***PSNR*** ↑	***SSIM*** ↑	***LPIPS*** ↓	***Mean Score (sec)***
DIV2K	23.04	0.636	0.321	163.4
DF2K	23.41	0.645	0.291	133.9
DF2K + OST	23.08	0.633	0.311	144.4

Source: The author (2022)

5.6 EXPERIMENTS FOR REAL-A-ESRGAN

5.6.1 Training Details

In this work, the A-ESRGAN was trained on the three datasets (DIV2K, DF2K and DF2K+OST). For better comparison with Real-ESRGAN, the authors followed the setting of training Real-ESRGAN and load the pre-trained Real-ESRNET to the generator of A-ESRGAN-Single. The training HR patch size is 256. The hyperparameters are a total batch size of 48 by using the Adam optimizer. Also for this case, the batch size was reduced from 48 to 8 due to computational limitations. The A-ESRGAN-Single is trained with a single attention U-Net discriminator for $400K$ iterations under $10E^{-4}$ rate. For the A-ESRGAN-Single, the weight for L_1 loss, perceptual loss, and GAN loss are $\{1, 1, 0.1\}$. The weight for GAN loss of D_{normal} and $D_{sampled}$ is 1, 1. This implementation, as the previous Real-ESRGAN model, was also based on BasicSR open-source toolbox. Table 8 shows the datasets, training time, parameters for generator and discriminator for the training of the A-ESRGAN model.

Table 8 – Training time consumed and parameters quantity for A-ESRGAN model.

Dataset	**Training time (hh:mm:ss)**	**Parameters G**	**Parameters D**
DIV2K	1 day, 18:32:01	16,697,987	5,399,044
DF2K	1 day, 16:49:27	16,697,987	5,399,044
DF2K + OST	1 day, 15:43:10	16,697,987	5,399,044

Source: The author (2022)

5.6.2 Testing Phase

After training the A-ESRGAN model, the generator and parameters are already trained with the weights ready for the inference process. Thus, it is used to generate 100 new SR images using the test set, with the corresponding ground truth images, from the DIV2K dataset, Consequently, the metrics (PSNR, SSIM, and LPIPS) are calculated. Table 9 shows the datasets, and the metrics for the trained A-ESRGAN model, including the mean score value for each evaluation.

Table 9 – A-ESRGAN Inference Results on the 100 test images in DIV2K dataset. Quantitative results for the A-ESRGAN model compared with the three selected datasets in which training was carried out.↑ and ↓ mean higher or lower is desired.

Dataset	***PSNR* ↑**	***SSIM* ↑**	***LPIPS* ↓**	***Mean Score (sec)***
DIV2K	23,71	0,659	0,309	154.8
DF2K	23,08	0,633	0,311	150.8
DF2K + OST	22,78	0,633	0,318	142.4

Source: The author (2022)

5.7 COMPARISON BETWEEN ALL EXPERIMENTS

Once all the experiments have been carried out, it is possible to create a global matrix where all the results obtained and interpreted using the respective metrics for the evaluation of the SR image-generating methods are shown. In all the results obtained, UR-SRGAN is the model with the best evaluation regarding the most representative metrics and is also the most used in competitions and the field of SR research. The objective of the experiments is to observe if the LPIPS metric can be improved in UR-SRGAN concerning the other three models used as a reference. In the experiments, it can be seen that the modification in the architecture of the GAN network discriminator and the addition of extra loss functions in the generator managed to reach the expectations of the proposed hypothesis.

Consequently, Table 10 displays the PSNR, SSIM, and LPIPS metrics that are calculated from results generated by the different methods proposed for the experiments. In this global comparison between all the trained models, it is seen that the model that achieved the best evaluation in the metrics was UR-SRGAN, having an average improvement compared to the other models in the PSNR, SSIM, and LPIPS metrics of 2.188, 0.0649. and 0.055, respectively. These results demonstrate that the implementation of U-net as a discriminator added to the other improvements at the level of data augmentation with CutMix and the estimation of images LR, gives superior results to the most recent models of SOTA in SR related to the BSR problem.

Table 10 – Global Inference Results among all trained SR models and datasets. Each value is the mean percentage of the sum of the 100 images tested under each model and dataset respectively.

SR MODEL	**DIV2K**			**DF2K**			**DF2K+OST**		
	PSNR ↑	***SSIM*** ↑	***LPIPS*** ↓	***PSNR*** ↑	***SSIM*** ↑	***LPIPS*** ↓	***PSNR*** ↑	***SSIM*** ↑	***LPIPS*** ↓
UR-SRGAN	**26.24**	**0.734**	**0.228**	**26.13**	**0.721**	**0.219**	**26.04**	**0.726**	**0.225**
RealSR	25.08	0.701	0.237	24.83	0.672	0.227	24.63	0.687	0.244
Real-ESRGAN	23.04	0.636	0.321	23.41	0.645	0.291	24.65	0.693	0.253
A-ESRGAN	23.71	0.659	0.309	23.08	0.633	0.311	22.78	0.633	0.318

Source: The author (2022)

Once all the experiments of the models of this work have been carried out: UR-SRGAN, of the comparative models in the three main datasets of images. A boxplot graph can be performed to see if UR-SRGAN truly shows an improvement in the metric selected after performing 100 experiments for each of the datasets of images and respective models. It can be seen that in the Figure 15 three analyzes are presented for each of the PSNR, SSIM and LPIPS metrics. Where the line inside the colored boxes represents the median of the 100 experiments carried out in each model and in the different datasets.

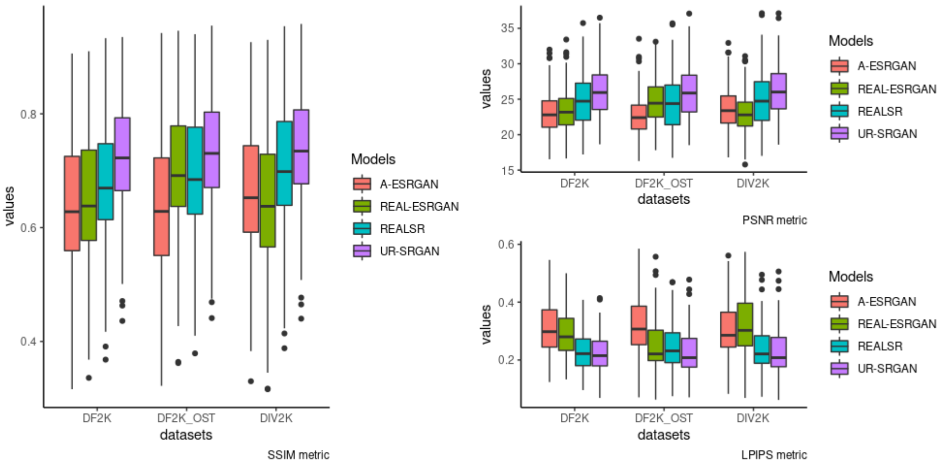
Since most of the distributions are symmetric (only some show a positively skewed distribution) and a minimal quantity of values exist out of the distribution that would be identified as outliers, it can be concluded that most of the result values were close to the median value. Therefore, if the model that its median is closer to the ideal value that is intended to be achieved with the experiments, this model will be the one that achieved the best results during the entire testing phase. In the graphic, it is clearly seen that UR-SRGAN outperforms the other counterpart models.

- SSIM metric: On the X axis (datasets) it represents the different image databases that were used in each one and the models in which they were executed are represented by 4 different colors. The Y axis (values) represents the distribution of the values obtained during the 100 evaluations in each of the validation images under the SSIM metric criterion in the interval [0 to 1], and value 1 is only reachable in the case of two identical sets of data and therefore indicates perfect structural similarity. Therefore, the median of the model that is closest to 1 will be the model that performed the best on the 100 test images.

- PSNR metric: On the X axis (datasets) it represents the different image databases that were used in each one and the models in which they were executed are represented by 4 different colors. The Y axis (values) represents the distribution of the values obtained during the 100 evaluations in each of the validation images under the PSNR metric criterion in the interval [0 to ∞], where the two compared images are identical, and thus the MSE is zero. In this case, the PSNR is ∞. Therefore, the median of the model that is closer to ∞ will be the model that performed the best on the 100 test images.

- LPIPS metric: On the X axis (datasets) it represents the different image databases that were used in each one and the models in which they were executed are represented by 4 different colors. The Y axis (values) represents the distribution of the values obtained during the 100 evaluations in each of the validation images under the LPIPS metric criterion where the closer the value is to zero, the better correlate to perceptual judgments. Therefore, the median of the model that is closest to 0 will be the model that performed the best on the 100 test images.

Figure 15 – Representative image of the four models making inference in 100 experiments (images) in three different databases.

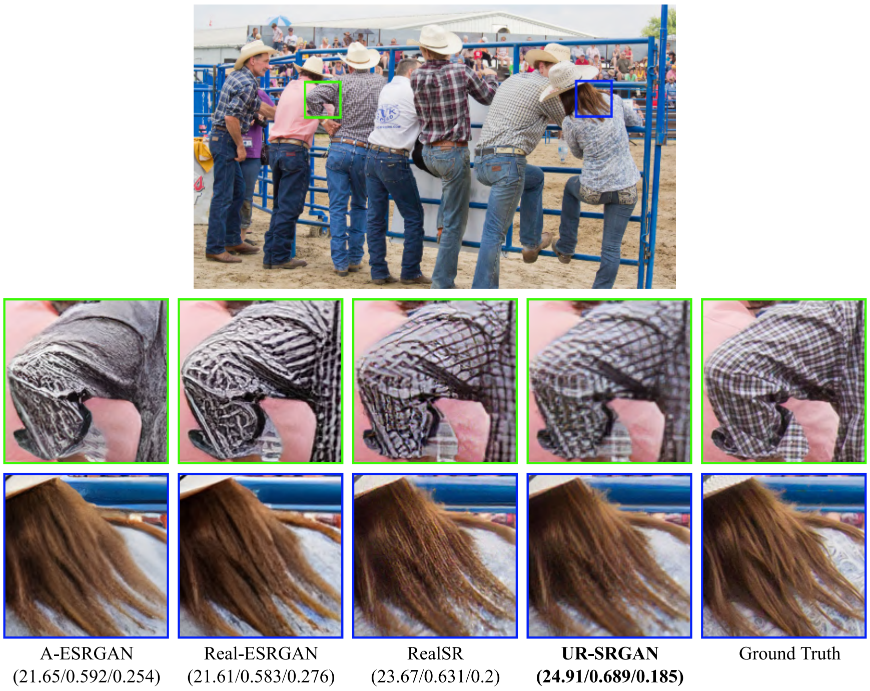


Source: The author (2022)

Next, some samples of the images generated by all the models after the experiments are presented. These images were generated in the following way: In the test dataset provided by DF2K and which has 100 images in LR and each of these has its respective GT image, an inference will be made of the four models that are used during this work. In the case of UR-SRGAN, the model that was trained in DF2K has been selected as the best option since it has better results according to the average of the three established metrics under the three dataset experiments. In the case of the counterpart models (RealSR, Real-ESRGAN, and A-ESRGAN) the trained model and the weights available by their respective authors are used, to compare the original version proposed and the present work in UR-SRGAN.

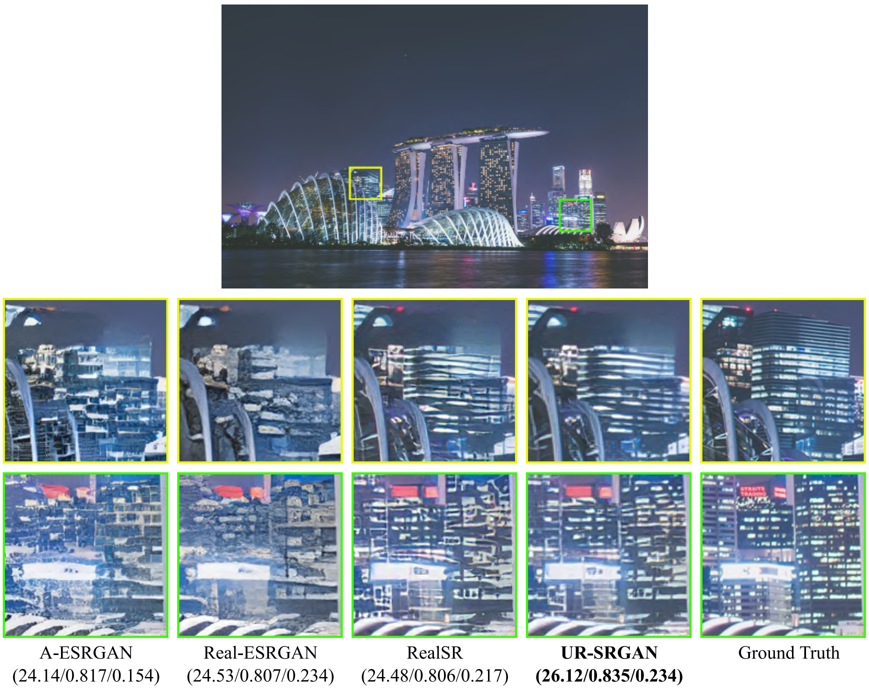
In each sample, the top image is the GT image and the bottom four images are a patch of the synthetic image generated by all the aforementioned models. The results that there exist important improvements with respect to other SOTA architectures, especially in the areas where there is greater detail of structures or lines, as seen in Figures 16, 17, and 21. There is also an improvement in details such as faces or body parts, as seen in Figures 22, and 20. Skin textures of animals, as seen in Figure 19, and symbols or characters, as seen in Figure 18. This model is much better suited to real-world images as other models generally have over-exposure artifacts to give better detail perception and look more real. It is important to emphasize that, in some cases, the models that are compared with UR-SRGAN can present images with a greater effect on image processing or eliminate possible artifacts. For example, in the sample 18 - the second row, in more recent models such as Real-ESRGAN and A-ESRGAN the details of the original design are completely lost. If there was no ground-truth image to compare the final result, it would be very well accepted since it is perceptibly correct.

Figure 16 – Input 804 (size: 510 × 300) from the DF2K LR dataset and its GT reference. Corresponding PSNR, SSIM, and LPIPS are shown in brackets. [4× upscaling].



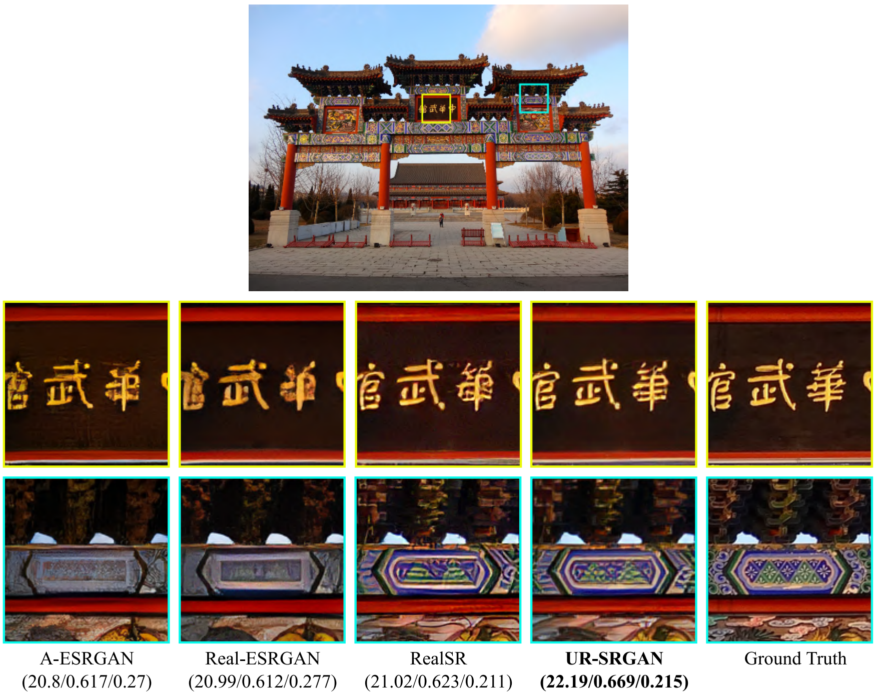
Source: The author (2022)

Figure 17 – Input 814 (size: 510 × 339) from the DF2K LR dataset and its GT reference. Corresponding PSNR, SSIM, and LPIPS are shown in brackets. [4× upscaling].



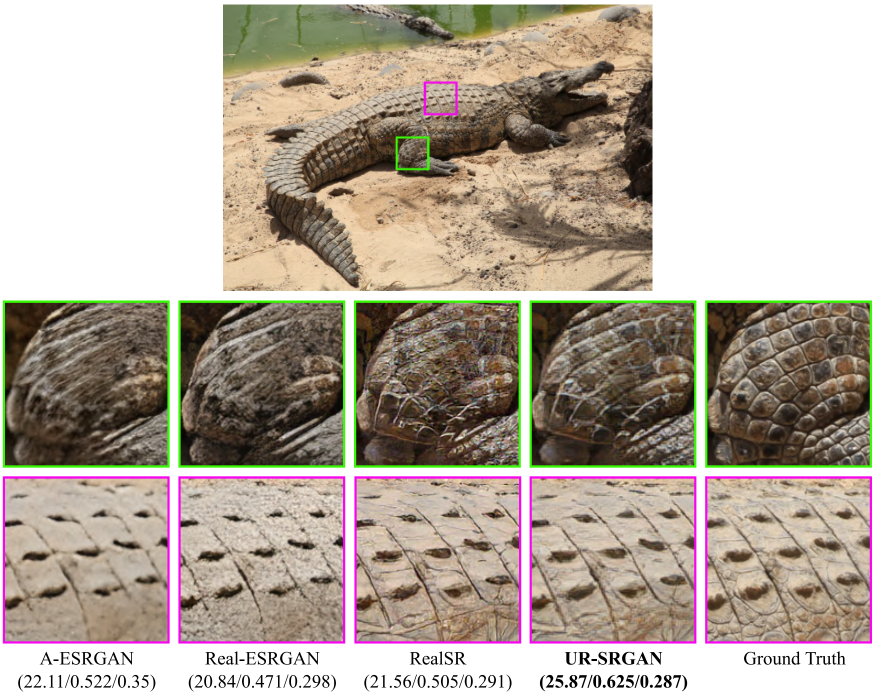
Source: The author (2022)

Figure 18 – Input 826 (size: 510 × 384) from the DF2K LR dataset and its GT reference. Corresponding PSNR, SSIM, and LPIPS are shown in brackets. [4× upscaling].



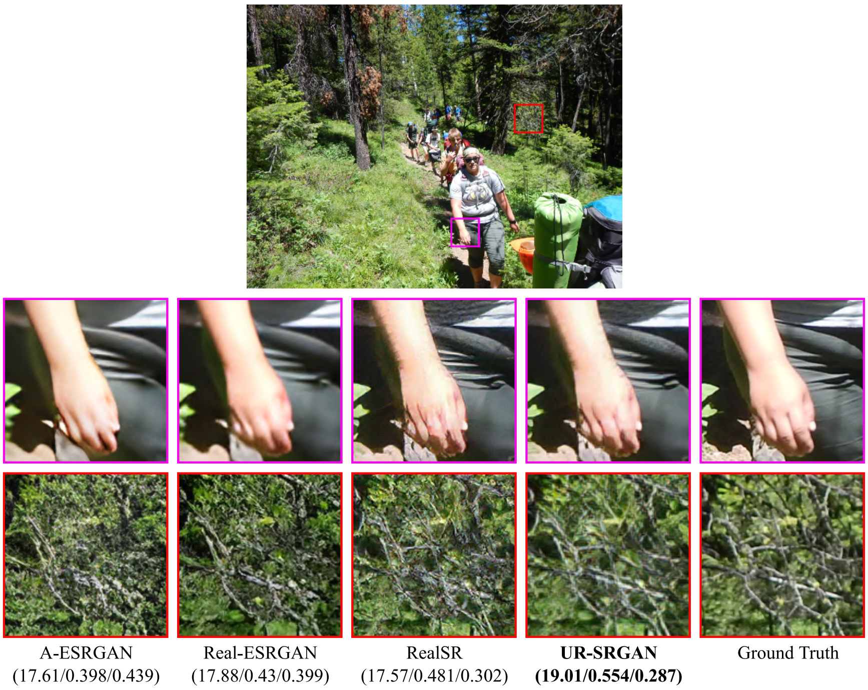
Source: The author (2022)

Figure 19 – Input 859 (size: 510 × 339) from the DF2K LR dataset and its GT reference. Corresponding PSNR, SSIM, and LPIPS are shown in brackets. [4× upscaling].



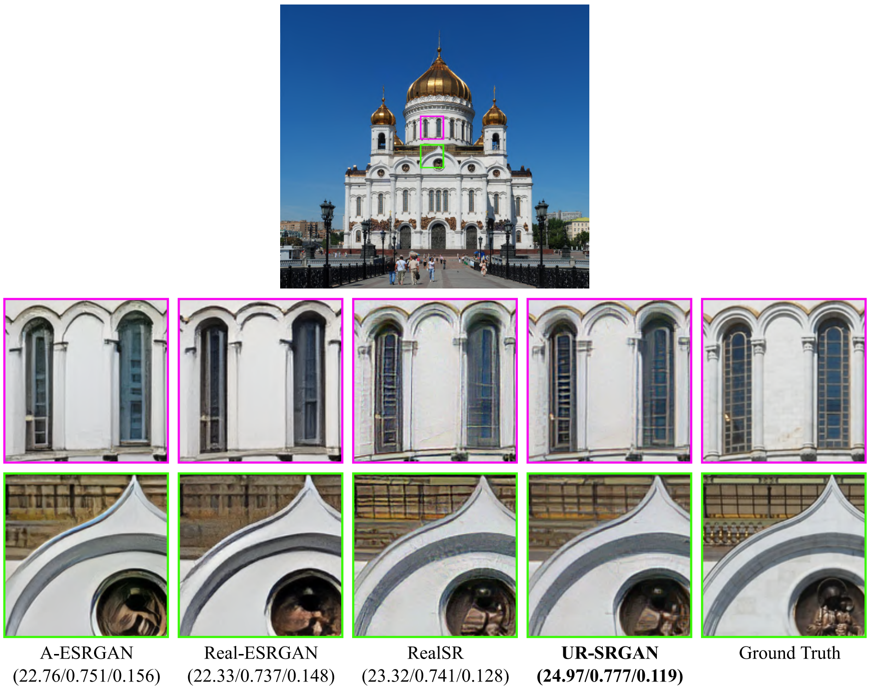
Source: The author (2022)

Figure 20 – Input 860 (size: 510 × 384) from the DF2K LR dataset and its GT reference. Corresponding PSNR, SSIM, and LPIPS are shown in brackets. [4× upscaling].



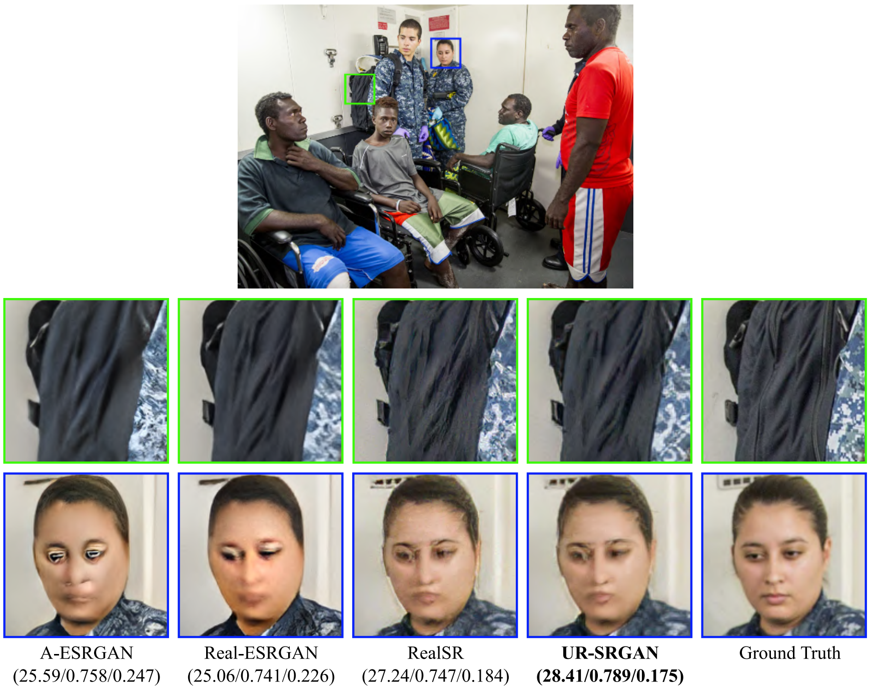
Source: The author (2022)

Figure 21 – Input 879 (size: 510 × 468) from the DF2K LR dataset and its GT reference. Corresponding PSNR, SSIM, and LPIPS are shown in brackets. [4× upscaling].



Source: The author (2022)

Figure 22 – Input 813 (size: 510 × 366) from the DF2K LR dataset and its GT reference. Corresponding PSNR, SSIM, and LPIPS are shown in brackets. [4× upscaling].



Source: The author (2022)

5.8 EVALUATION ON REAL-WORLD IMAGES

To better evaluate the new proposed architecture of UR-SRGAN, a test was performed on the 100 LR images found in the DPED dataset. For this, the UR-SRGAN model that obtained the best average results was used, which was the experiment executed on the DF2K dataset. The rest of the models used the weights trained originally by the authors to perform a better comparison. Since there is no Ground-truth image with which to compare the metrics PSNR, SSIM, LPIPS, the NIQE metric will be used, which is a completely blind image quality analyzer that only makes use of measurable deviations from statistical regularities observed in natural images, without training on human-rated distorted images, and, indeed, without any exposure to distorted images.

A lower NIQE value indicates better perceptual quality. During the 100 new experiments, it can be seen in Table 11 that UR-SRGAN obtains good results with respect to the DPED dataset, however, lower values are recorded for the results of A-ESRGAN. Nevertheless, UR-SRGAN achieves the second-best qualification, taking into account that, as will be appreciated in the example images, the architecture proposed in this work presents better perceptual results and look natural.

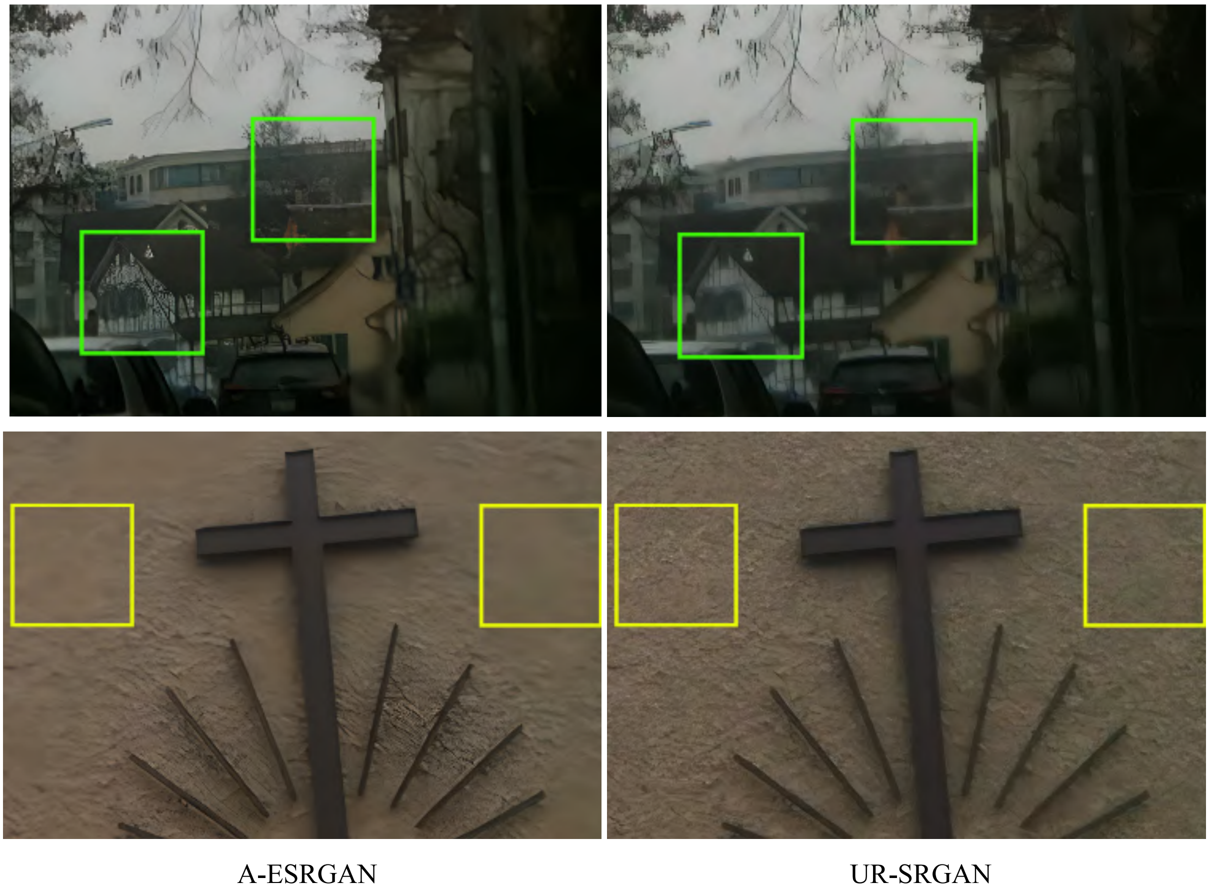
Table 11 – Inference Results using DPED validation dataset over all SR trained models. NIQE Mean score value after 100 experiments over DPED dataset real-world images. ↓ means lower is desired.

	A-ESRGAN	**Real-ESRGAN**	**RealSR**	**UR-SRGAN**
NIQE ↓	**4.152**	5.082	4.835	4.672

Source: The author (2022)

The A-ESRGAN model presents satisfactory results in relation to the NIQE metrics, however many of the synthetic images that this model returns contain noticeable artifacts that disturb the entire image even when other regions of the image are presented with higher details. It is possible to see some examples in the Figure 23 below:

Figure 23 – Some generated images where the artifacts of the A-ESRGAN model are notorious with respect to the same synthetic HR images retrieved from the UR-SRGAN model.

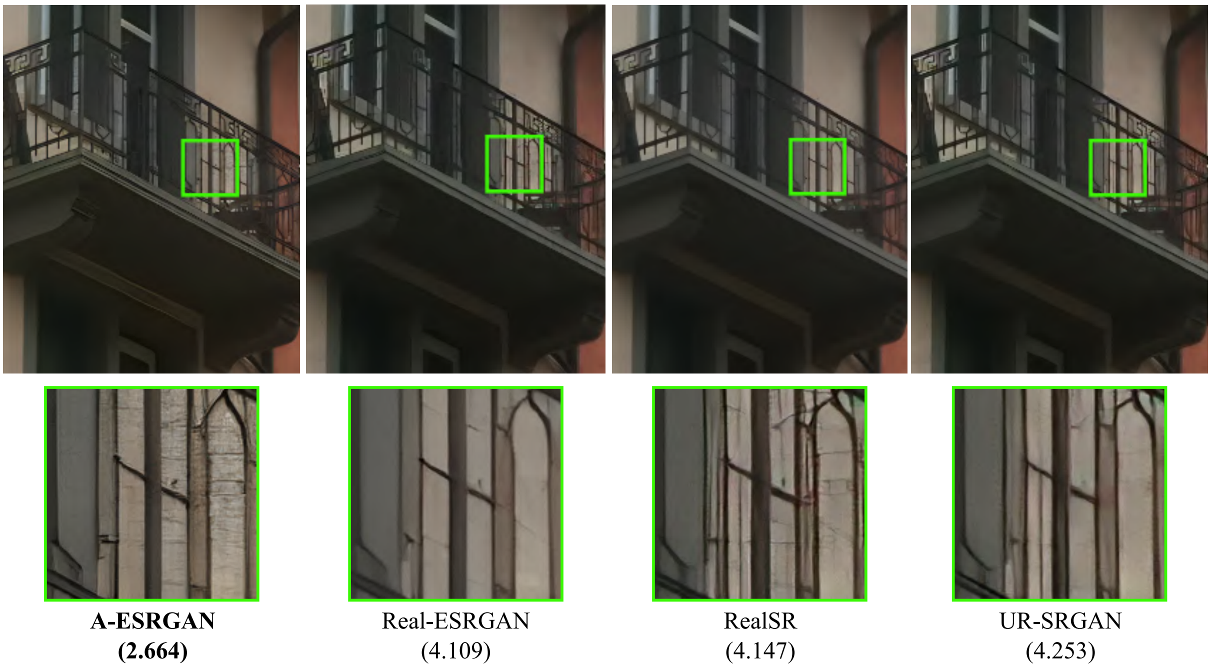


Source: The author (2022)

In context, it might be preferable to have a model that generates an image that presents a complete global result and highlights local details more moderately than images generated with discontinuous and speckled local structures or images with inconsistent geometric and structural patterns.

Finally, to have a visualization of all the models, additional samples will be shown on the DPED real-world validation images to see the quality of these results and even be able to perceive that UR-SRGAN maintains its consistency in generating images with details better adapted to images of unknown degradation or obtained from different sources. devices or cameras. In Figures 24,25 can be appreciated the improvement in details for sharper lines and less noisy images for UR-SRGAN results. In Figures, 25, and 27, the resulting images are more realistic than the other generated images.

Figure 24 – Input 00017 (size: 512 × 256) from the DPED LR dataset without GT reference. Corresponding NIQE is shown in brackets. [4× upscaling].



A-ESRGAN
(2.664)

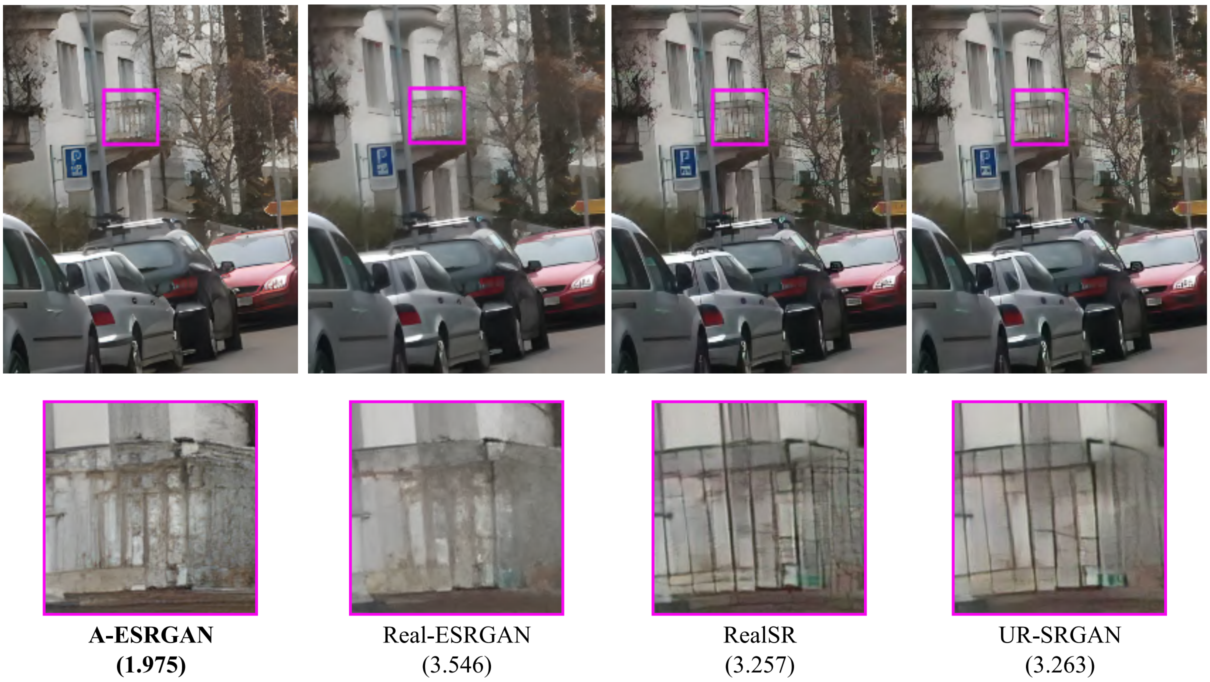
Real-ESRGAN
(4.109)

RealSR
(4.147)

UR-SRGAN
(4.253)

Source: The author (2022)

Figure 25 – Input 00029 (size: 512 × 256) from the DPED LR dataset without GT reference. Corresponding NIQE is shown in brackets. [4× upscaling].)



A-ESRGAN
(1.975)

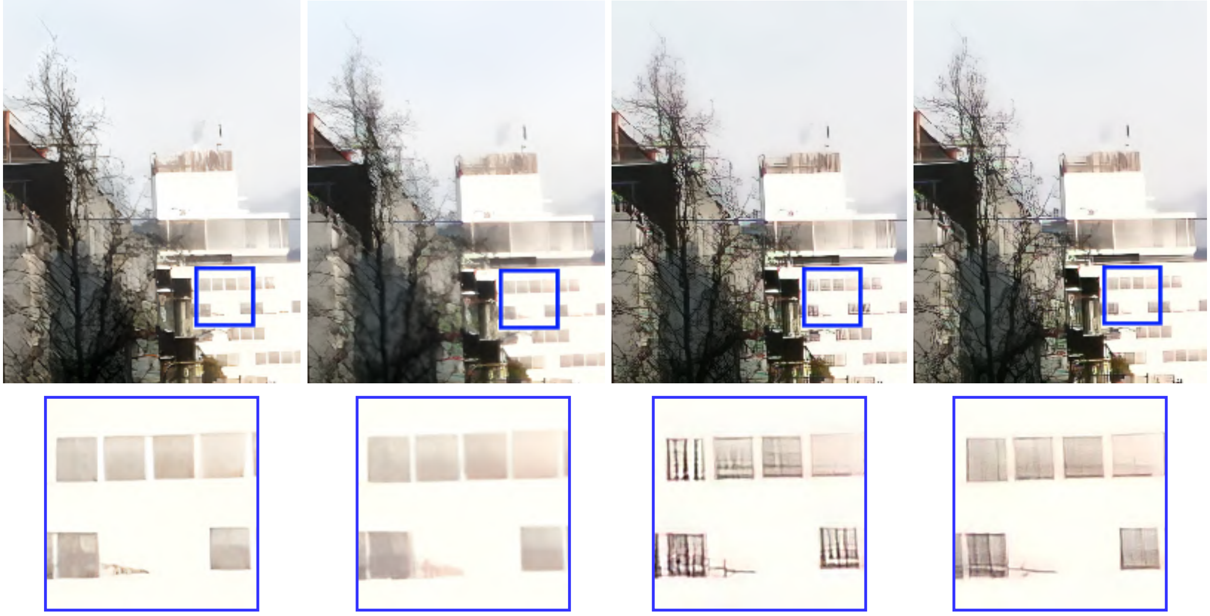
Real-ESRGAN
(3.546)

RealSR
(3.257)

UR-SRGAN
(3.263)

Source: The author (2022)

Figure 26 – Input 00049 (size: 512 × 256) from the DPED LR dataset without GT reference. Corresponding NIQE is shown in brackets. [4× upscaling].



A-ESRGAN
(2.233)

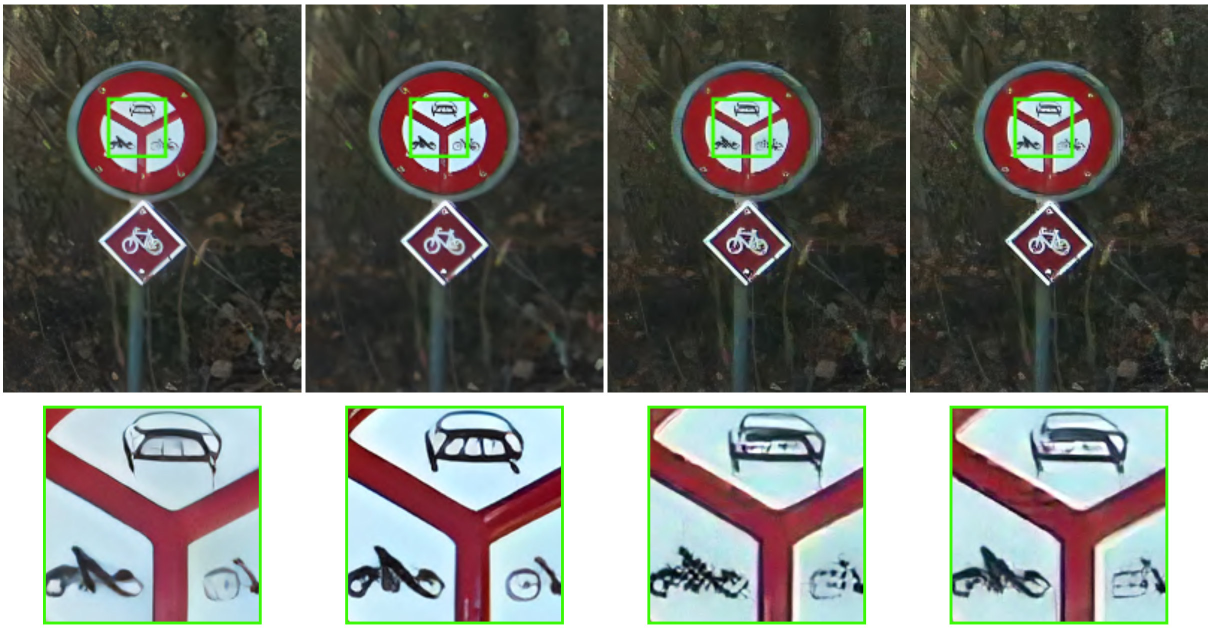
Real-ESRGAN
(2.619)

RealSR
(5.016)

UR-SRGAN
(2.863)

Source: The author (2022)

Figure 27 – Input 00093 (size: 512 × 256) from the DPED LR dataset without GT reference. Corresponding NIQE is shown in brackets. [4× upscaling].



A-ESRGAN
(3.029)

Real-ESRGAN
(6.312)

RealSR
(3.672)

UR-SRGAN
(2.889)

Source: The author (2022)

6 CONCLUSIONS AND FUTURE WORK

6.1 SUMMARY OF RESULTS

In this work, a novel UR-SRGAN for high-resolution image generation from low-resolution images with unknown degradation (BSR) was proposed. The U-Net architecture has become a common generator structure in many domains of GAN. It is for these reasons, along with its popularity, that it was chosen to use the U-Net architecture in the UR-SRGAN framework as well. The introduced U-Net-based discriminator is not only capable of providing per-pixel feedback to the denoising network but also focuses on the global structure. The main objective of this scheme was to add a U-Net-based discriminator, which can enhance the edge information and alleviate generated images with discontinued and mottled local structures or images with incoherent geometric and structural patterns.

At the pre-processing stage, an estimation of the degradation and noise extraction presented by real-world images was implemented to create a pool of kernels and noises. Once these kernels and noises had been obtained, the next step was to downsample the images of the training dataset with a bicubic process so that the noise that could exist is eliminated. In addition, by having LR images, the kernels and noise obtained from the real images were applied randomly so that the model could train more adequately and manage other images of the real world that it has never seen. In addition, it was also examined that the CutMix technique can boost the training of the discriminator with data augmentation. The employment of these CutMix images is included for consistency regularization, penalizing per-pixel inconsistent predictions of the discriminator under the CutMix transformations. To give more focus to the perceptual results, two new loss functions related to LPIPS and Feature Matching were added to the GAN network generator module.

Several experiments were performed to analyze the U-Net-based GAN structure that would improve at generating high-resolution images from low-resolution with unknown degradations. First, the UR-SRGAN was trained with the same datasets and hyperparameters that the other three SOTA models used as the baseline for comparison; RealSR, Real-ESRGAN, and A-ESRGAN SR models. These datasets were the DIV2K, DF2K, and DF2K + OST, and for the testing phase, two datasets were used; DF2K and DPED. Following the experimental phase, each of these architectures had to be trained on the three selected datasets to compare the metrics of the results at a more exact level. All models were trained with the same hyperparameters originally used by the authors to make a fairer comparison and evidence if UR-SRGAN equaled or improved the metrics acquired for the high-resolution images. All these models varied in time difference when being trained due to the variations in parameters that each one presents. Only one model had to be modified in batch size, which was for the Real-ESRGAN models due to the limitation of the computer on which it was trained.

At the testing phase, all models in the two datasets were evaluated with the metrics PSNR, SSIM, LPIPS and NIQE. The number of images that both datasets have is 100 images. After running the inference on the four super-resolution models, including UR-SRGAN, an analysis of the results at a quantitative and qualitative level was possible. About the metrics, it was observed that there is a significant improvement in the mean acquired between the metrics PSNR, SSIM, LPIPS of 2.188, 0.0649 and 0.055, respectively. This improvement is important if we consider that the UR-SRGAN model was compared with the most relevant models in GAN architectures currently focused on super-resolution for images with complex degradation. In addition, when comparing the images visually, you can see the improvement in the context of the image and in focused details such as greater detail of structures such as the scales or fur of animals, buildings with more natural characteristics and less geometric incoherence, sharper lines and people's faces without many distortions, getting closer to the GT images. It is important to mention that in the second DPED dataset of images taken by an iPhone 3GS, the UR-SRGAN model got second place concerning the other models in the metric NIQE. However, at a qualitative level, it outperforms the model that has the best score in this metric since the opposite model generates some distorted images with noise that disturb the global context of the image.

Finally, the UR-SRGAN demonstrated that including a U-Net discriminator had a notable improvement in providing both global and local feedback to the GAN generator. In addition, three other factors were relevant to obtain higher metric evaluations; the CutMix data augmentation, adding the LPIPS and Feature Match loss functions and adding a pre-processing method to generate LR images for the training using kernel estimation and noise injection strategies. After several experiments, the HR images generated by UR-SRGAN outperform the most recent SOTA SRGAN models. Therefore, this architecture is an important contribution to investigating the BSR problem for generating satisfactory images from LR real-world images in which source and distributions are unknown.

6.2 LIMITATIONS

It is assumed that there are some images that are generated by UR-SRGAN that are outperformed by other models. For example, Real-ESRGAN has a better performance when carrying out a face enhancement because it has a GFPGAN module incorporated, however, when carrying out this improvement, many times the model hallucinates artifacts placed on any object in the image that has the form of a face, although, in reality, it is not the case. On the other hand, this proposal is based on adding a U-Net architecture as a discriminator, however after this work began, other models appeared with similar proposals in GAN.

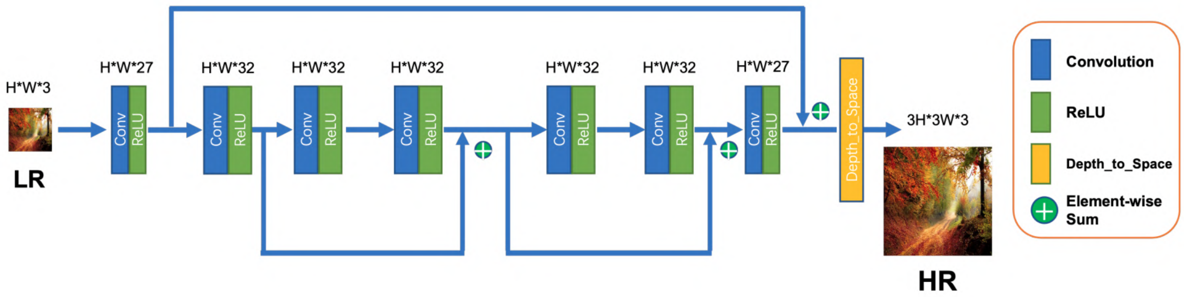
For this reason, a comparison is made with the main ones to compare the results between UR-SRGAN and the other proposals. Finally, alternatively to GAN networks, there are other proposals for SR based on auto-encoders that may be much superior to GAN networks in the future, however, just because something is old does not mean it is inferior. The GAN proposals in SR, as the present work exposes UR-SRGAN, give good results to obtain HR images with an adequate and accessible computational cost for their training.

Finally, in the experiments, only images [x4] times larger than the input image were generated. It is required to work on the architecture to add modules that allow higher resolution e.g., [x8, x16]. It is an essential requirement to evaluate the resulting image at a quality level.

6.3 FUTURE WORK

First, an interesting contribution in the area of SISR is to bring the UR-SRGAN model to mobile devices or edge devices. A reference work is (CAI et al., 2022) that designs a real-time ISR model for mobile devices able to deal with a wide range of degradations in real-world scenarios. It is implied that several constraints prevent CNN deployment on mobile devices: a restricted amount of RAM, many common CNN operators not supported, limited FLOPs, and power consumption requirements for mobile devices. Therefore, they utilize an entire 8-bit QAT strategy and design neural architecture using hardware friendly operations, their whole architecture is shown in Figure 28 setting portable meta-operators and time-consuming meta-operators.

Figure 28 – An illustration in color of the proposed InnoPeak mobileSR for mobile devices.



Source: (CAI et al., 2022)

Another ideal approach to follow is (AYAZOGLU, 2021) that proposes a hardware (Synaptics Dolphin NPU) limitation-aware, extremely lightweight quantization robust real-time SR network (XLSR). They applied root modules to the SISR problem using

Clipped ReLU at the last layer of the network to make the model uint8 quantization robust, achieving a better balance between reconstruction quality and runtime. In addition, their proposal won the Mobile AI 2021 Real-Time SISR Challenge.

Based on these previous works, the UR-SRGAN can follow a similar structure adapted for mobile devices. The primary operations would be divided into four categories: tensor operator nodes (Concatenation and Summation), convolution nodes (Convolution and Transposed Convolution), activation nodes (ReLU), and resize nodes (Convolution, Transposed Convolution, depth to space and space to depth) which would mainly consist of four parts: Shallow feature extraction part, which transfers the LR image to feature space. Deep feature extraction part, which learns high-level information and restores details such as edges, and textures. Skips connection part, and the reconstruction part, which maps feature space to HR image.

REFERENCES

AGUSTSSON, E.; TIMOFTE, R. Ntire 2017 challenge on single image super-resolution: Dataset and study. In: *Proceedings of the IEEE conference on computer vision and pattern recognition workshops*. [S.l.: s.n.], 2017. p. 126–135.

AHN, N.; KANG, B.; SOHN, K.-A. Fast, accurate, and lightweight super-resolution with cascading residual network. In: *Proceedings of the European conference on computer vision (ECCV)*. [S.l.: s.n.], 2018. p. 252–268.

AYAZOGLU, M. Extremely lightweight quantization robust real-time single-image super resolution for mobile devices. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. [S.l.: s.n.], 2021. p. 2472–2479.

BELL-KLIGLER, S.; SHOCHER, A.; IRANI, M. Blind super-resolution kernel estimation using an internal-gan. *Advances in Neural Information Processing Systems*, v. 32, 2019.

CAI, J.; MENG, Z.; DING, J.; HO, C. M. Real-time super-resolution for real-world images on mobile devices. *arXiv preprint arXiv:2206.01777*, 2022.

CHEN, J.; CHEN, J.; CHAO, H.; YANG, M. Image blind denoising with generative adversarial network based noise modeling. In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. [S.l.: s.n.], 2018. p. 3155–3164.

CHOI, J.-S.; KIM, M. A deep convolutional neural network with selection units for super-resolution. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*. [S.l.: s.n.], 2017.

COLLIER, E.; DUFFY, K.; GANGULY, S.; MADANGUIT, G.; KALIA, S.; SHREEKANT, G.; NEMANI, R.; MICHAELIS, A.; LI, S.; GANGULY, A. et al. Progressively growing generative adversarial networks for high resolution semantic segmentation of satellite images. In: IEEE. *2018 IEEE International Conference on Data Mining Workshops (ICDMW)*. [S.l.], 2018. p. 763–769.

CORNILLèRE, V.; DJELOUAH, A.; YIFAN, W.; SORKINE-HORNUNG, O.; SCHROERS, C. Blind image super-resolution with spatially variant degradations. *ACM Trans. Graph.*, Association for Computing Machinery, New York, NY, USA, v. 38, n. 6, nov 2019. ISSN 0730-0301. Disponível em: <https://doi.org/10.1145/3355089.3356575>.

CRESWELL, A.; WHITE, T.; DUMOULIN, V.; ARULKUMARAN, K.; SENGUPTA, B.; BHARATH, A. A. Generative adversarial networks: An overview. *IEEE signal processing magazine*, IEEE, v. 35, n. 1, p. 53–65, 2018.

DONG, C.; LOY, C. C.; HE, K.; TANG, X. Image super-resolution using deep convolutional networks. *IEEE transactions on pattern analysis and machine intelligence*, IEEE, v. 38, n. 2, p. 295–307, 2015.

DONG, C.; LOY, C. C.; TANG, X. Accelerating the super-resolution convolutional neural network. In: SPRINGER. *European conference on computer vision*. [S.l.], 2016. p. 391–407.

DONG, X.; LEI, Y.; WANG, T.; THOMAS, M.; TANG, L.; CURRAN, W. J.; LIU, T.; YANG, X. Automatic multiorgan segmentation in thorax ct images using u-net-gan. *Medical physics*, Wiley Online Library, v. 46, n. 5, p. 2157–2168, 2019.

ECABERT, O.; PETERS, J.; SCHRAMM, H.; LORENZ, C.; BERG, J. von; WALKER, M. J.; VEMBAR, M.; OLSZEWSKI, M. E.; SUBRAMANYAN, K.; LAVI, G. et al. Automatic model-based segmentation of the heart in ct images. *IEEE transactions on medical imaging*, IEEE, v. 27, n. 9, p. 1189–1201, 2008.

EL-SAMIE, F. E. A.; HADHOUD, M. M.; EL-KHAMY, S. E. *Image Super-Resolution and Applications*. London, England: CRC Press, 2019.

GOODFELLOW, I.; POUGET-ABADIE, J.; MIRZA, M.; XU, B.; WARDE-FARLEY, D.; OZAIR, S.; COURVILLE, A.; BENGIO, Y. Generative adversarial nets. *Advances in neural information processing systems*, v. 27, 2014.

GU, J.; LU, H.; ZUO, W.; DONG, C. Blind super-resolution with iterative kernel correction. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. [S.l.: s.n.], 2019.

HUANG, H.; LIN, L.; TONG, R.; HU, H.; ZHANG, Q.; IWAMOTO, Y.; HAN, X.; CHEN, Y.-W.; WU, J. Unet 3+: A full-scale connected unet for medical image segmentation. In: IEEE. *ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. [S.l.], 2020. p. 1055–1059.

HUANG, Y.; LI, S.; WANG, L.; TAN, T. et al. Unfolding the alternating optimization for blind super resolution. *Advances in Neural Information Processing Systems*, v. 33, p. 5632–5643, 2020.

HUANG, Z.; ZHANG, J.; ZHANG, Y.; SHAN, H. Du-gan: Generative adversarial networks with dual-domain u-net-based discriminators for low-dose ct denoising. *IEEE Transactions on Instrumentation and Measurement*, IEEE, v. 71, p. 1–12, 2021.

HUI, Z.; LI, J.; WANG, X.; GAO, X. Learning the non-differentiable optimization for blind super-resolution. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. [S.l.: s.n.], 2021. p. 2093–2102.

IBTEHAZ, N.; RAHMAN, M. S. Multiresunet: Rethinking the u-net architecture for multimodal biomedical image segmentation. *Neural networks*, Elsevier, v. 121, p. 74–87, 2020.

IGNATOV, A.; KOBYSHEV, N.; TIMOFTE, R.; VANHOEY, K.; GOOL, L. V. Dslr-quality photos on mobile devices with deep convolutional networks. In: *Proceedings of the IEEE International Conference on Computer Vision*. [S.l.: s.n.], 2017. p. 3277–3285.

ISOLA, P.; ZHU, J.-Y.; ZHOU, T.; EFROS, A. A. Image-to-image translation with conditional adversarial networks. In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. [S.l.: s.n.], 2017. p. 1125–1134.

JI, X.; CAO, Y.; TAI, Y.; WANG, C.; LI, J.; HUANG, F. Real-world super-resolution via kernel estimation and noise injection. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*. [S.l.: s.n.], 2020.

JI, X.; CAO, Y.; TAI, Y.; WANG, C.; LI, J.; HUANG, F. Real-world super-resolution via kernel estimation and noise injection. In: *proceedings of the IEEE/CVF conference on computer vision and pattern recognition workshops*. [S.l.: s.n.], 2020. p. 466–467.

JO, Y.; YANG, S.; KIM, S. J. Investigating loss functions for extreme super-resolution. In: *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition workshops*. [S.l.: s.n.], 2020. p. 424–425.

JOHNSON, J.; ALAHI, A.; FEI-FEI, L. Perceptual losses for real-time style transfer and super-resolution. In: SPRINGER. *European conference on computer vision*. [S.l.], 2016. p. 694–711.

KERFOOT, E.; CLOUGH, J.; OKSUZ, I.; LEE, J.; KING, A. P.; SCHNABEL, J. A. Left-ventricle quantification using residual u-net. In: SPRINGER. *International Workshop on Statistical Atlases and Computational Models of the Heart*. [S.l.], 2018. p. 371–380.

KIM, J.; LEE, J. K.; LEE, K. M. Accurate image super-resolution using very deep convolutional networks. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. [S.l.: s.n.], 2016.

KIM, J.; LEE, J. K.; LEE, K. M. Accurate image super-resolution using very deep convolutional networks. In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. [S.l.: s.n.], 2016. p. 1646–1654.

KIM, J.; LEE, J. K.; LEE, K. M. Deeply-recursive convolutional network for image super-resolution. In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. [S.l.: s.n.], 2016. p. 1637–1645.

KIM, J.; LEE, S. Deep learning of human visual sensitivity in image quality assessment framework. In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. [S.l.: s.n.], 2017. p. 1676–1684.

KIM, S. Y.; SIM, H.; KIM, M. Koalanet: Blind super-resolution using kernel-oriented adaptive local adjustment. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. [S.l.: s.n.], 2021. p. 10611–10620.

LAI, W.-S.; HUANG, J.-B.; AHUJA, N.; YANG, M.-H. Deep laplacian pyramid networks for fast and accurate super-resolution. In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. [S.l.: s.n.], 2017. p. 624–632.

LEDIG, C.; THEIS, L.; HUSZÁR, F.; CABALLERO, J.; CUNNINGHAM, A.; ACOSTA, A.; AITKEN, A.; TEJANI, A.; TOTZ, J.; WANG, Z. et al. Photo-realistic single image super-resolution using a generative adversarial network. In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. [S.l.: s.n.], 2017. p. 4681–4690.

LEE, O.-Y.; SHIN, Y.-H.; KIM, J.-O. Multi-perspective discriminators-based generative adversarial network for image super resolution. *IEEE Access*, IEEE, v. 7, p. 136496–136510, 2019.

LIM, B.; SON, S.; KIM, H.; NAH, S.; LEE, K. M. Enhanced deep residual networks for single image super-resolution. In: *2017 IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*. [S.l.: s.n.], 2017. p. 1132–1140.

LIU, A.; LIU, Y.; GU, J.; QIAO, Y.; DONG, C. Blind image super-resolution: A survey and beyond. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, IEEE, 2022.

LOU, A.; GUAN, S.; LOEW, M. Dc-unet: rethinking the u-net architecture with dual channel efficient cnn for medical image segmentation. In: SPIE. *Medical Imaging 2021: Image Processing.* [S.l.], 2021. v. 11596, p. 758–768.

LUGMAYR, A.; DANELLJAN, M.; TIMOFTE, R. Ntire 2020 challenge on real-world image super-resolution: Methods and results. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops.* [S.l.: s.n.], 2020. p. 494–495.

LUO, z.; HUANG, Y.; LI, S.; WANG, L.; TAN, T. Unfolding the alternating optimization for blind super resolution. In: LAROCHELLE, H.; RANZATO, M.; HADSELL, R.; BALCAN, M.; LIN, H. (Ed.). *Advances in Neural Information Processing Systems.* Curran Associates, Inc., 2020. v. 33, p. 5632–5643. Disponível em: <https://proceedings.neurips.cc/paper/2020/file/3d2d8ccb37df977cb6d9da15b76c3f3a-Paper.pdf>.

MA, C.; RAO, Y.; LU, J.; ZHOU, J. Structure-preserving image super-resolution. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, p. 1–1, 2021.

MA, C.; YANG, C.-Y.; YANG, X.; YANG, M.-H. Learning a no-reference quality metric for single-image super-resolution. *Computer Vision and Image Understanding*, Elsevier, v. 158, p. 1–16, 2017.

MILANFAR, P. (Ed.). *Super-Resolution Imaging.* Boca Raton, FL: CRC Press, 2009. (Digital Imaging and Computer Vision).

MITTAL, A.; SOUNDARARAJAN, R.; BOVIK, A. C. Making a “completely blind” image quality analyzer. *IEEE Signal processing letters*, IEEE, v. 20, n. 3, p. 209–212, 2012.

RAMWALA, O. A.; PAUNWALA, C. N.; PAUNWALA, M. C. Image de-raining for driver assistance systems using u-net based gan. In: IEEE. *2019 IEEE International Conference on Signal Processing, Information, Communication & Systems (SPICSCON).* [S.l.], 2019. p. 23–26.

REN, H.; KHERADMAND, A.; EL-KHAMY, M.; WANG, S.; BAI, D.; LEE, J. Real-world super-resolution using generative adversarial networks. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) Workshops.* [S.l.: s.n.], 2020.

RONNEBERGER, O.; FISCHER, P.; BROX, T. U-net: Convolutional networks for biomedical image segmentation. In: SPRINGER. *International Conference on Medical image computing and computer-assisted intervention.* [S.l.], 2015. p. 234–241.

SCHONFELD, E.; SCHIELE, B.; KHOREVA, A. A u-net based discriminator for generative adversarial networks. In: *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition.* [S.l.: s.n.], 2020. p. 8207–8216.

SHI, W.; CABALLERO, J.; HUSZÁR, F.; TOTZ, J.; AITKEN, A. P.; BISHOP, R.; RUECKERT, D.; WANG, Z. Real-time single image and video super-resolution using an efficient sub-pixel convolutional neural network. In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. [S.l.: s.n.], 2016. p. 1874–1883.

SIMONYAN, K.; ZISSERMAN, A. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014.

TAI, Y.; YANG, J.; LIU, X. Image super-resolution via deep recursive residual network. In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. [S.l.: s.n.], 2017. p. 3147–3155.

TALEBI, H.; MILANFAR, P. Nima: Neural image assessment. *IEEE transactions on image processing*, IEEE, v. 27, n. 8, p. 3998–4011, 2018.

TIMOFTE, R.; AGUSTSSON, E.; GOOL, L. V.; YANG, M.-H.; ZHANG, L. Ntire 2017 challenge on single image super-resolution: Methods and results. In: *Proceedings of the IEEE conference on computer vision and pattern recognition workshops*. [S.l.: s.n.], 2017. p. 114–125.

WANG, C.; XU, C.; WANG, C.; TAO, D. Perceptual adversarial networks for image-to-image transformation. *IEEE Transactions on Image Processing*, IEEE, v. 27, n. 8, p. 4066–4079, 2018.

WANG, L.; WANG, Y.; DONG, X.; XU, Q.; YANG, J.; AN, W.; GUO, Y. Unsupervised degradation representation learning for blind super-resolution. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. [S.l.: s.n.], 2021. p. 10581–10590.

WANG, X.; XIE, L.; DONG, C.; SHAN, Y. Real-esrgan: Training real-world blind super-resolution with pure synthetic data. In: *Proceedings of the IEEE/CVF International Conference on Computer Vision*. [S.l.: s.n.], 2021. p. 1905–1914.

WANG, X.; YU, K.; CHAN, K. C.; DONG, C.; LOY, C. C. *BasicSR: Open Source Image and Video Restoration Toolbox*. 2018. <https://github.com/xinntao/BasicSR>.

WANG, X.; YU, K.; WU, S.; GU, J.; LIU, Y.; DONG, C.; QIAO, Y.; LOY, C. C. Esrgan: Enhanced super-resolution generative adversarial networks. In: *Proceedings of the European Conference on Computer Vision (ECCV) Workshops*. [S.l.: s.n.], 2018.

WANG, X.; YU, K.; WU, S.; GU, J.; LIU, Y.; DONG, C.; QIAO, Y.; LOY, C. C. Esrgan: Enhanced super-resolution generative adversarial networks. In: *Proceedings of the European Conference on Computer Vision (ECCV) Workshops*. [S.l.: s.n.], 2018.

WANG, Z.; BOVIK, A. C.; SHEIKH, H. R.; SIMONCELLI, E. P. Image quality assessment: from error visibility to structural similarity. *IEEE transactions on image processing*, IEEE, v. 13, n. 4, p. 600–612, 2004.

WANG, Z.; CHEN, J.; HOI, S. C. Deep learning for image super-resolution: A survey. *IEEE transactions on pattern analysis and machine intelligence*, IEEE, v. 43, n. 10, p. 3365–3387, 2020.

WEI, Z.; HUANG, Y.; CHEN, Y.; ZHENG, C.; GAO, J. *A-ESRGAN: Training Real-World Blind Super-Resolution with Attention U-Net Discriminators*. arXiv, 2021. Disponível em: <https://arxiv.org/abs/2112.10046>.

WEI, Z.; HUANG, Y.; CHEN, Y.; ZHENG, C.; GAO, J. *A-ESRGAN: Training Real-World Blind Super-Resolution with Attention U-Net Discriminators*. arXiv, 2021. Disponível em: <https://arxiv.org/abs/2112.10046>.

WENG, Y.; ZHOU, T.; LI, Y.; QIU, X. Nas-unet: Neural architecture search for medical image segmentation. *IEEE Access*, IEEE, v. 7, p. 44247–44257, 2019.

XU, Y.-S.; TSENG, S.-Y. R.; TSENG, Y.; KUO, H.-K.; TSAI, Y.-M. Unified dynamic convolutional network for super-resolution with variational degradations. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. [S.l.: s.n.], 2020.

YI, X.; BABYN, P. Sharpness-aware low-dose ct denoising using conditional generative adversarial network. *Journal of digital imaging*, Springer, v. 31, n. 5, p. 655–669, 2018.

YOSINSKI, J.; CLUNE, J.; BENGIO, Y.; LIPSON, H. How transferable are features in deep neural networks? *Advances in neural information processing systems*, v. 27, 2014.

YUE, Z.; ZHAO, Q.; XIE, J.; ZHANG, L.; MENG, D.; WONG, K.-Y. K. Blind image super-resolution with elaborate degradation modeling on noise and kernel. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. [S.l.: s.n.], 2022. p. 2128–2138.

YUN, S.; HAN, D.; OH, S. J.; CHUN, S.; CHOE, J.; YOO, Y. Cutmix: Regularization strategy to train strong classifiers with localizable features. In: *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*. [S.l.: s.n.], 2019.

ZHANG, H.; CISSÉ, M.; DAUPHIN, Y. N.; LOPEZ-PAZ, D. mixup: Beyond empirical risk minimization. *CoRR*, abs/1710.09412, 2017. Disponível em: <http://arxiv.org/abs/1710.09412>.

ZHANG, K.; GOOL, L. V.; TIMOFTE, R. Deep unfolding network for image super-resolution. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. [S.l.: s.n.], 2020.

ZHANG, K.; LIANG, J.; GOOL, L. V.; TIMOFTE, R. Designing a practical degradation model for deep blind image super-resolution. In: *Proceedings of the IEEE/CVF International Conference on Computer Vision*. [S.l.: s.n.], 2021. p. 4791–4800.

ZHANG, K.; ZUO, W.; ZHANG, L. Learning a single convolutional super-resolution network for multiple degradations. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. [S.l.: s.n.], 2018.

ZHANG, K.; ZUO, W.; ZHANG, L. Deep plug-and-play super-resolution for arbitrary blur kernels. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. [S.l.: s.n.], 2019.

ZHANG, R.; ISOLA, P.; EFROS, A. A.; SHECHTMAN, E.; WANG, O. The unreasonable effectiveness of deep features as a perceptual metric. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. [S.l.: s.n.], 2018.

ZHANG, W.; SHI, G.; LIU, Y.; DONG, C.; WU, X.-M. A closer look at blind super-resolution: Degradation models, baselines, and performance upper bounds. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. [S.l.: s.n.], 2022. p. 527–536.

ZHANG, Y.; LI, K.; LI, K.; WANG, L.; ZHONG, B.; FU, Y. Image super-resolution using very deep residual channel attention networks. In: *Proceedings of the European conference on computer vision (ECCV)*. [S.l.: s.n.], 2018. p. 286–301.

ZHOU, R.; SUSSTRUNK, S. Kernel modeling super-resolution on real low-resolution images. In: *Proceedings of the IEEE/CVF International Conference on Computer Vision*. [S.l.: s.n.], 2019. p. 2433–2443.

ZHOU, R.; SüSSTRUNK, S. Kernel modeling super-resolution on real low-resolution images. In: *2019 IEEE/CVF International Conference on Computer Vision (ICCV)*. [S.l.: s.n.], 2019. p. 2433–2443.

ZHOU, Z.; SIDDIQUEE, M. M. R.; TAJBAKHSH, N.; LIANG, J. Unet++: A nested u-net architecture for medical image segmentation. In: *Deep learning in medical image analysis and multimodal learning for clinical decision support*. [S.l.]: Springer, 2018. p. 3–11.

APPENDIX A – U-NET DISCRIMINATOR IMPLEMENTATION

Listing A.1 – Programming language code Python with Pytorch Framework: U-Net Discriminator Implementation, source: The author (2022)

```

### U-Net Discriminator ###
# Residual block for the discriminator
class DiscriminatorBlock(nn.Module):
    def __init__(self, in_channels, out_channels, which_conv=nn.Conv2d, which_bn=nn.
        BatchNorm2d, wide=True,
                preactivation=True, activation=nn.LeakyReLU(0.1, inplace=False),
                    downsample=nn.AvgPool2d(2, stride=2)):
        super(DiscriminatorBlock, self).__init__()
        self.in_channels, self.out_channels = in_channels, out_channels
        self.hidden_channels = self.out_channels if wide else self.in_channels
        self.which_conv, self.which_bn = which_conv, which_bn
        self.preactivation = preactivation
        self.activation = activation
        self.downsample = downsample
        # Conv layers
        self.conv1 = self.which_conv(self.in_channels, self.hidden_channels,
            kernel_size=3, padding=1)
        self.conv2 = self.which_conv(self.hidden_channels, self.out_channels,
            kernel_size=3, padding=1)
        self.learnable_sc = True if (in_channels != out_channels) or downsample else
             False
        if self.learnable_sc:
            self.conv_sc = self.which_conv(in_channels, out_channels,
                                            kernel_size=1, padding=0)
        self.bn1 = self.which_bn(self.hidden_channels)
        self.bn2 = self.which_bn(out_channels)

    def forward(self, x):
        if self.preactivation:
            h = self.activation(x)
        else:
            h = x
        h = self.bn1(self.conv1(h))
        if self.downsample:
            h = self.downsample(h)
        return h

class GeneratorBlock(nn.Module):
    def __init__(self, in_channels, out_channels,
                which_conv=nn.Conv2d, which_bn=nn.BatchNorm2d, activation=nn.
                    LeakyReLU(0.1, inplace=False),
                upsample=nn.Upsample(scale_factor=2, mode='nearest')):
        super(GeneratorBlock, self).__init__()
        self.in_channels, self.out_channels = in_channels, out_channels
        self.which_conv, self.which_bn = which_conv, which_bn
        self.activation = activation
        self.upsample = upsample
        # Conv layers
        self.conv1 = self.which_conv(self.in_channels, self.out_channels,
            kernel_size=3, padding=1)
        self.conv2 = self.which_conv(self.out_channels, self.out_channels,
            kernel_size=3, padding=1)
        self.learnable_sc = in_channels != out_channels or upsample
        if self.learnable_sc:
            self.conv_sc = self.which_conv(in_channels, out_channels,
                                            kernel_size=1, padding=0)
        # Batchnorm layers
        self.bn1 = self.which_bn(out_channels)
        self.bn2 = self.which_bn(out_channels)
        # upsample layers
        self.upsample = upsample

    def forward(self, x):
        h = self.activation(x)
        if self.upsample:
            h = self.upsample(h)
        h = self.bn1(self.conv1(h))
        return h

class UnetDiscriminator(torch.nn.Module):
    def __init__(self):
        super(UnetDiscriminator, self).__init__()

        self.enc_b1 = DiscriminatorBlock(3, 64, preactivation=False)
        self.enc_b2 = DiscriminatorBlock(64, 128)
        self.enc_b3 = DiscriminatorBlock(128, 192)
        self.enc_b4 = DiscriminatorBlock(192, 256)
        self.enc_b5 = DiscriminatorBlock(256, 320)
        self.enc_b6 = DiscriminatorBlock(320, 384)

        self.enc_out = nn.Conv2d(384, 1, kernel_size=1, padding=0)

        self.dec_b1 = GeneratorBlock(384, 320)
        self.dec_b2 = GeneratorBlock(320*2, 256)
        self.dec_b3 = GeneratorBlock(256*2, 192)
        self.dec_b4 = GeneratorBlock(192*2, 128)
        self.dec_b5 = GeneratorBlock(128*2, 64)
        self.dec_b6 = GeneratorBlock(64*2, 32)

        self.dec_out = nn.Conv2d(32, 1, kernel_size=1, padding=0)

        # Init weights
        for m in self.modules():
            classname = m.__class__.__name__
            if classname.lower().find('conv') != -1:
                nn.init.kaiming_normal(m.weight)
                nn.init.constant(m.bias, 0)
            elif classname.find('bn') != -1:
                m.weight.data.normal_(1.0, 0.02)
                m.bias.data.fill_(0)

    def forward(self, x):
        e1 = self.enc_b1(x)
        e2 = self.enc_b2(e1)
        e3 = self.enc_b3(e2)
        e4 = self.enc_b4(e3)
        e5 = self.enc_b5(e4)
        e6 = self.enc_b6(e5)

        e_out = self.enc_out(F.leaky_relu(e6, 0.1))

        d1 = self.dec_b1(e6)
        d2 = self.dec_b2(torch.cat([d1, e5], 1))
        d3 = self.dec_b3(torch.cat([d2, e4], 1))
        d4 = self.dec_b4(torch.cat([d3, e3], 1))
        d5 = self.dec_b5(torch.cat([d4, e2], 1))
        d6 = self.dec_b6(torch.cat([d5, e1], 1))

        d_out = self.dec_out(F.leaky_relu(d6, 0.1))

        return e_out, d_out, [e1,e2,e3,e4,e5,e6], [d1,d2,d3,d4,d5,d6]
```