

UNIVERSIDADE FEDERAL DO RIO GRANDE DO SUL  
INSTITUTO DE INFORMÁTICA  
CURSO DE CIÊNCIA DA COMPUTAÇÃO

SURYALALL DOS SANTOS RAHOO

## **Image Restoration with Neural Networks**

Undergraduate Thesis presented in partial fulfillment  
of the requirements for the degree of Bachelor of  
Computer Science

Advisor: Prof. Dr. Manuel Menezes de Oliveira Neto

Porto Alegre  
2023

UNIVERSIDADE FEDERAL DO RIO GRANDE DO SUL

Reitor: Prof. Carlos André Bulhões Mendes

Vice-Reitora: Prof<sup>ª</sup> Patricia Helena Lucas Pranke

Pró-Reitoria de Ensino (Graduação e Pós-Graduação): Prof<sup>ª</sup> Cíntia Inês Boll

Diretora do Instituto de Informática: Prof<sup>ª</sup> Carla Maria Dal Sasso Freitas

Coordenador do Curso de Ciência da Computação: Prof. Marcelo Walter

Bibliotecário-Chefe do Instituto de Informática: Alexsander Borges Ribeiro

## **ACKNOWLEDGEMENTS**

I would like to thank my parents who have encouraged me to complete my studies. A special thanks to my mother Sara who has given me unconditional support and has always been there for me. Thanks to the rest of my family as well who has been loving towards me ever since I was little. I would also like to show appreciation to Cleyde who has helped me in tough moments and still helps me with invaluable advice. Thanks to Prof. Manuel who has advised me throughout this undergraduate thesis and who has encouraged me to study deep learning. Thanks to everyone else who also had a positive impact in my life. Finally, I would like thank myself for believing in me, for my work, and for the thousands of hours I have invested in my own education.

## ABSTRACT

This undergraduate thesis condenses an 8 month-long study on deep image restoration which is a fast-growing field with many real-world applications. It includes a discussion on traditional image restoration, a review of 26 datasets, and a survey comprising 5 different image restoration tasks, i.e., deblurring, super-resolution, non-blind restoration, face-restoration, and video restoration. For each task, 2 neural network-based methods are described and compared. Moreover, this thesis discusses 3 experimental image restoration techniques which we have developed, specifically: Gradient Descent Deconvolution (GDDec) for non-blind deblurring, Super-Resolution Residual U-Net (SRRUNet) for super-resolution, and Our Non-Blind which is a non-blind framework. Finally, this thesis provides various research directions which contemplate each method discussed.

**Keywords:** Image Restoration. Deep Learning. Neural Networks. Survey.

# Restauração de Imagens utilizando Redes Neurais

## RESUMO

Este trabalho de graduação condensa um estudo de 8 meses sobre o processo de restauração de imagens utilizando técnicas de aprendizagem profunda: um campo em rápido crescimento com muitas aplicações no mundo real. O trabalho inclui uma discussão sobre a área de restauração de imagens, uma revisão de 26 bases de dados utilizadas para teste e treinamento de modelos e um estudo compreendendo 5 tarefas de restauração de imagens: remoção de borramento, super-resolução, restauração não cega, restauração de rostos e restauração de vídeos. Para cada tarefa, 2 métodos que utilizam redes neurais são descritos e comparados. Além disso, esta tese discute 3 técnicas experimentais de restauração de imagens que elaboramos, especificamente: Gradient Descent Deconvolution (GDDec) para remoção não cega de borramento, Super-Resolution Residual U-Net (SRRUNet) para super-resolução, e Our Non-Blind para restauração não cega. Finalmente, esta tese fornece várias direções de pesquisa que contemplam cada método discutido.

**Palavras-chave:** Restauração de Imagens. Aprendizado Profundo. Rede Neurais.

## LIST OF FIGURES

Figure 3.1 – Degradation modeling where $x = y + z$ .....	15
Figure 3.2 – (a) Original residual network building block. (b) ResBlock. (c) Res FFT-Conv Block. CONV, BN and RELU stand for Convolutional, Batch Normalization, and ReLu layers, respectively. .....	16
Figure 3.3 – MIMO-UNet backbone. Shallow Convolutional Module (SCM) extracts features and Asymmetric Feature Fusion (AFF) merges multi-scale features. ....	17
Figure 3.4 – Overview of Uformer-B. Both Input and Output Projections are $3 \times 3$ convolutional layers while the former adds a LeakyReLU activation.....	18
Figure 3.5 – LeWin Transformer block. LN is Layer Normalization and W-MSA is Window-based Multi-head Self-Attention. ....	19
Figure 3.6 – Comparison of DeepRFT+ and Uformer-B applied to synthetic uniform blur removal...	20
Figure 3.7 – Comparison of DeepRFT+ and Uformer-B applied to in the wild non-uniform blur removal.....	21
Figure 4.1 – Example of blocking artifacts in two different feature maps of SwinIR. ....	23
Figure 4.2 – HAT architecture. ....	23
Figure 4.3 – LR input on the left side, some of its (normalized) VGG features in the middle, and corresponding OOE map on the right side. ....	25
Figure 4.4 – A single Dense Block. ....	25
Figure 4.5 – Super-resolution comparison with scale factor $sf = 4$ .....	26
Figure 4.6 – Super-resolution comparison with scale factor $sf = 16$ . Metrics cannot be computed for images of different sizes the reason why they are not shown. ....	26
Figure 5.1 – Main components of Our Non-Blind. ....	28
Figure 5.2 – DRUNet's architecture. ....	29
Figure 5.3 – Results of GDDec and results of removing ringing artifacts using the recursive filter (RF) with $(\sigma_s, \sigma_r) = (60, 0.5)$ .....	31
Figure 5.4 – $69 \times 69$ Blur kernel estimated by the method of Pan et al. (2016).....	31
Figure 5.5 – Results of GDDec when using an estimated blur kernel.....	32
Figure 5.6 – Before and after fine-tuning SRRUNet.....	33
Figure 5.7 – Noise-free low resolution image. ....	35
Figure 5.8 – Extremely noisy image. ....	35
Figure 5.9 – Gradient descent vs. USRNet's data module. Input is a noise-free image with large blur kernel and a scale factor of 1.....	36
Figure 5.10 – DRUNet vs. ResUNet. Input is blur-free image with 50% gaussian noise and a scale factor of 1. ....	36
Figure 6.1 – StyleGAN overview.....	38
Figure 6.2 – Normalized feature maps for each layer of GPEN. The encoder fmap at layer $k$ is denoted by $E_k$ . Likewise for the decoder.....	39
Figure 6.3 – Downscaling images leads to loss of information. When the resulting face image is very small (e.g., $16 \times 16$ ) there is no way of telling whose face it is. ....	40
Figure 6.4 – Computer simulated degradation leading to a $16 \times 16$ image.....	41
Figure 6.5 – In the wild BFR.....	42
Figure 6.6 – Image-to-image translation using GPEN. ....	43
Figure 7.1 – BasicVSR++ framework.....	44
Figure 7.2 – VRT framework.....	45
Figure 7.3 – Experiments on Vid4. ....	46
Figure 7.4 – Experiments on REDS4. ....	47

## LIST OF ABBREVIATIONS AND ACRONYMS

BFR	Blind Face Restoration
BN	Batch Normalization
CNN	Convolutional Neural Network
DeepRFT	Deep Residual Fourier Transformation
DNN	Deep Neural Network
DRUNet	Denoising Residual U-Net
GAN	Generative Adversarial Network
GDDec	Gradient Descent Deconvolution
GPEN	GAN Prior Embedded Network
HAT	Hybrid Attention Transformer
HQ	High Quality
HR	High Resolution
LPIPS	Learned Perceptual Image Patch Similarity
LQ	Low Quality
LR	Low Resolution
MLP	Multilayer Perceptron
PSNR	Peak Signal to Noise Ratio
PULSE	Photo Upsampling via Latent Space Exploration
ReLU	Rectified Linear Unit
ResBlock	Residual Block
RGB	Red Green Blue
SF	Scale Factor
SISR	Single-Image Super-Resolution
SOTA	State-of-the-Art
SROOE	Super-Resolution Optimal Objective Estimation
SRRUNet	Super-Resolution Residual U-Net
SSIM	Structural Similarity
Uformer	U-Shaped Transformer
USRNet	Unfolding Super-Resolution Network
VRT	Video Restoration Transformer
VSR	Video Super-Resolution

## CONTENTS

<b>1 INTRODUCTION</b> .....	<b>9</b>
<b>2 BACKGROUND</b> .....	<b>10</b>
<b>2.1 Image Restoration</b> .....	<b>10</b>
<b>2.2 Metrics</b> .....	<b>12</b>
<b>2.3 Datasets</b> .....	<b>12</b>
<b>3 DEBLURRING</b> .....	<b>15</b>
<b>3.1 DeepRFT</b> .....	<b>15</b>
<b>3.2 Uformer</b> .....	<b>17</b>
<b>3.3 Uformer vs. DeepRFT</b> .....	<b>19</b>
<b>4 SUPER-RESOLUTION</b> .....	<b>22</b>
<b>4.1 HAT</b> .....	<b>22</b>
<b>4.2 SROOE</b> .....	<b>24</b>
<b>4.3 SROOE vs. HAT</b> .....	<b>26</b>
<b>5 NON-BLIND RESTORATION</b> .....	<b>28</b>
<b>5.1 Our Non-Blind</b> .....	<b>28</b>
5.1.1 GDDec .....	29
5.1.2 SRRUNet .....	32
<b>5.2 USRNet</b> .....	<b>34</b>
<b>5.3 USRNet vs. Our Non-Blind</b> .....	<b>34</b>
<b>6 FACE RESTORATION</b> .....	<b>38</b>
<b>6.1 GPEN</b> .....	<b>38</b>
<b>6.2 PULSE</b> .....	<b>40</b>
<b>6.3 PULSE vs. GPEN</b> .....	<b>41</b>
<b>7 VIDEO RESTORATION</b> .....	<b>44</b>
<b>7.1 BasicVSR++</b> .....	<b>44</b>
<b>7.2 VRT</b> .....	<b>45</b>
<b>7.3 VRT vs. BasicVSR++</b> .....	<b>45</b>
<b>8 CONCLUSION</b> .....	<b>48</b>
<b>8.1 Research Directions</b> .....	<b>49</b>
<b>REFERENCES</b> .....	<b>51</b>



# 1 INTRODUCTION

Wrapping one's mind around the mechanics of neural networks is serious business. While their fast evaluation comes from efficient implementation of neural layers in machine learning libraries, their efficacy comes from having had their weights and biases optimized according to some loss function.

With the aim of acquiring understanding and intuition on how, why, and when neural networks can be used for the purposes of image restoration, this work discusses existing methods that entail at least one neural component. Those include general modules such as residual blocks, transformers, U-Nets, and GANs. On top of that, it covers varying image restoration tasks with the additional goal of conceiving a broader picture of the field. To that end, five tasks ranging from deblurring to video restoration are addressed and, for each task, two selected methods are discussed at length. The selection is based on (i) which methods produced the best results at the time this text was written, and (ii) availability of both source code and pretrained models.

The remainder of this thesis is organized as follows: Chapter 2 begins by contextualizing the actual field of image restoration. Next, metrics are discussed followed by a review of 26 datasets commonly used in deep image restoration. Chapter 3 discusses kernel agnostic methods which are capable of both uniform and non-uniform deblurring. Next, Chapter 4 deals with single image super-resolution (SISR) methods whose aim is to recover a high-resolution (HR) image from a low-resolution (LR) image. Subsequently, Chapter 5 returns to the blur kernel paradigm within a unified degradation model comprising blur, downsampling and noise, and presents “Our Non-Blind” which is a framework composed of different methods including “GDDec” and “SRRUNet”. Then, Chapter 6 analyzes techniques which leverage implicit priors learned by generative networks to better recover images of specific domains, e.g., human faces. Lastly, Chapter 7 describes strategies adopted by state-of-the-art video-restoration methods. Finally, Chapter 8 summarizes key conclusions which can be drawn from this study.

## 2 BACKGROUND

This chapter provides an overview on image restoration and introduces relevant concepts which are further discussed in subsequent chapters.

### 2.1 Image Restoration

Convolutions permeate image processing and neural networks. A convolution  $\otimes$  is an operation that processes an input in a sliding window fashion, applying a kernel to a patch of the corresponding kernel size, at each window position. Mathematically, a blurry image  $y$  can be modeled as the convolution output of its original sharp counterpart  $x$  with a blur kernel  $k$ , that is,

$$y = (x \otimes k). \quad (2.1)$$

This special type of kernel, also known as **Point Spread Function (PSF)**, is characterized by (i) having non-negative values, and (ii) the sum of its elements being equal to one. If this kernel is known, the blurry image can be deconvolved in a so-called "non-blind" manner.

Inverse filtering is the simplest form of non-blind deconvolution. It is based on the convolution theorem which states that

$$y = (x \otimes k) = \mathcal{F}^{-1}(\mathcal{F}(x) * \mathcal{F}(k)), \quad (2.2)$$

where  $\mathcal{F}$  and  $\mathcal{F}^{-1}$  are Fourier Transform and its inverse,  $*$  is a pointwise multiplication operator, and  $\mathcal{F}(k)$  is the **Optical Transfer Function (OTF)**. Thus, inverse filtering computes

$$x = \mathcal{F}^{-1}(\mathcal{F}(y)/\mathcal{F}(k)). \quad (2.3)$$

However, because blurring is often accompanied by other degradations such as random additive noise and/or compression which are difficult to accurately estimate, inverse filtering alone easily leads to unrecognizable outputs as it is prone to noise amplification, depending on the values of  $\mathcal{F}(k)$ . Wiener Filter (WIENER, 1964) also builds on the convolution theorem, but tries to minimize noise impact. Nevertheless, it requires the power spectra of both noise  $n$  and original image  $x$ . While spectrally white noise has constant power spectrum, the power spectrum can be approximated using the  $1/f$  power law, from a set of natural images (POULI; REINHARD; CUNNINGHAM, 2013). The PSF, also typically unknown, can be estimated.

Shan, Jia and Agarwala (2008) proposed a unified model of both blind and non-blind deconvolution which iteratively alternated between refining blur kernel and restoring image.

Upon convergence, it would produce high quality deblurred results while avoiding deconvolution artifacts and was comparable to techniques which required additional input photographs or additional hardware. Afterwards, similar approaches were proposed. Pan et al. (2013) and Pan et al. (2016) developed faster algorithms which would converge even in larger blur settings achieving state-of-the-art results then.

Additive noise  $n$  can be parameterized by  $n = \sigma N$ , where  $\sigma$  is the noise level (or standard deviation), and  $N$  is a sampled distribution (e.g., Gaussian or Poissonian). The median filter is an elementary noise reduction approach. It is a nonlinear filter which produces, for each output pixel, the median of input pixel values under a sliding window centered at the reference input pixel. The result is a smoothed image with less visible noise, but the computation becomes inefficient as the patch size increases (due to the computation of the medians). In addition, it performs poorly when dealing with large amounts of Gaussian noise. Nonetheless, the median filter is a good choice when dealing with salt and pepper noise.

Resolution refers to the amount of detail in an image. During downsampling for instance, photos inevitably lose detail. However, this detail cannot be recovered by simply upsampling the image using resizing algorithms such as bicubic, bilinear, or nearest-neighbor. Instead, the resulting low-resolution images are pixelated. Although simple geometric shapes such as lines and curves can be vectorized thus replacing jagged lines with smooth lines, real life photos require more sophisticated approaches to actually improve image quality. Finally, the resizing of both image height and width can be parameterized by a scale factor  $sf$  which can be broken into horizontal and vertical components ( $sf_x, sf_y$ ). During upscaling,  $sf$  is a multiplier and during downscaling it is a divisor.

Advances within the field of deep learning led to the development of new image restoration methods which are radically different from traditional ones. In deblurring, for instance, CNNs no longer assumed any restricted blur kernel model and, as such, could directly recover sharp images without kernel estimation (ZHANG et al., 2022) as well as better deal with non-uniform deblurring (NAH; KIM; LEE, 2017). Deep denoisers, for the first time, completely removed noise in extreme cases of noise level up to 200 (ZHANG et al., 2021). With specific training data, networks could be tailored for specific ends such as deblurring photos of human faces (YASARLA; PERAZZI; PATEL, 2020) and recreating historical photos as if they were taken by modern cameras (LUO et al., 2021).

Notably, deep image restoration serves as an exciting field which contributes to the seemingly endless barrage of scientific papers. Moreover, the methods described in the

following chapters have several applications which include: removing blur caused by camera shake or motion blur from photos shot using conventional cameras and smartphones; aiding other computer vision tasks through a preprocessing step, such as in medical imaging, satellite imaging, face recognition in surveillance footage, and other forms of object recognition; and simply enhancing overall image and video quality.

## 2.2 Metrics

By convention, image restoration scientists resort to metrics for a quantitative evaluation of their own method(s) with regard to those of others. Likewise, Chapter 3 adopts the commonly used PSNR $\uparrow$  (GONZALEZ; WOODS, 2006) along with SSIM $\uparrow$  (WANG et al., 2004), and LPIPS $\downarrow$  (ZHANG et al., 2018), where  $\uparrow/\downarrow$  arrow indicates whether a better performance correlates to having a higher/lower metric value. These metrics compare the original clean image unseen by the method, and the final estimated output. PSNR $\uparrow$  and SSIM $\uparrow$  have simple mathematical expressions:

$$PSNR = 10 \log_{10} \frac{I_{max}^2}{MSE}, \quad (2.4)$$

where  $I_{max}$  is the maximum signal extent, e.g.,  $I_{max} = 255$  for eight-bit images, and  $MSE$  is the mean squared error measured between ground truth image and output image; and

$$SSIM(x, y) = \frac{2\mu_x\mu_y + C_1}{\mu_x^2 + \mu_y^2 + C_1} \cdot \frac{2\sigma_{xy} + C_2}{\sigma_x^2 + \sigma_y^2 + C_2}, \quad (2.5)$$

where  $x$  and  $y$  are image patches of equal size with height=width,  $\mu_x$  is the mean of  $x$ ,  $\sigma_x^2$  is the variance of  $x$ ,  $\sigma_{xy}$  is the covariance of  $x$  and  $y$ , and both  $C_1$  and  $C_2$  have default values and are used to stabilize the division. Finally, LPIPS $\downarrow$  compares activations of the image pair using some pretrained network, e.g., VGG (SIMONYAN; ZISSERMAN, 2015).

## 2.3 Datasets

This section reviews the different datasets used by each model described in chapters 3, 4, 5, 6, and 7. Following the description, the list of models which trains on these datasets is enclosed in brackets.

**AFHQ** (CHOI et al., 2020): 16,130 images of animal faces. All images have size 512 $\times$ 512. [SRRUNet]

**BSD68** (MARTIN et al., 2001): 68 images for image denoising benchmarks.

**BSD100** (MARTIN et al., 2001): 100 test images of natural scenes.

**BSD400** (CHEN; POCK, 2017): 400 images. [DRUNet]

**CelebA-HQ** (KARRAS et al., 2017): 30,000 face images of size 1024×1024.

**DF2K** (LIM et al., 2017): 3,450 images which are actually the result of merging DIV2K with Flickr2K. [HAT; SROOE]

**DIV2K** (AGUSTSSON; TIMOFTE, 2017): 800 images. [DRUNet; HAT; SROOE; USRNet]

**DPDD** (ABUOLAIM; BROWN, 2020): 2,000 images for deblurring tasks. [DeepRFT]

**FFHQ** (KARRAS; LAINE; AILA, 2018): Similar to CelebA-HQ, FFHQ contains 70,000 face images of size 1024×1024. [SRRUNet; GPEN]

**Flickr1024** (WANG et al., 2019): 1,024 image pairs of varying sizes. [SRRUNet]

**Flickr2K** (TIMOFTE et al., 2017): 2,650 images from Flickr. [DRUNet; HAT; USRNet]

**General100** (DONG; LOY; TANG, 2016): 100 bmp-format images with no compression and of size ranging from 710×704 to 131×112. Proposed for SR training.

**GOPRO** (NAH; KIM; LEE, 2017): 33 video clips with frames of size 1280×720. It contains a total of 3,214 pairs of blurry and sharp images. [DeepRFT; SRRUNet; Uformer; VRT]

**HIDE** (SHEN et al., 2019): 8,422 blurry-sharp image pairs (6,397 for training and 2,025 for testing).

**ImageNet** (DENG et al., 2009): 14,197,122 annotated images organized by the semantic hierarchy of WordNet. Often used for pre-training. [HAT]

**Manga109** (MATSUI et al., 2017): 109 Japanese manga images of size 827×1170. [SRRUNet]

**RealBlur** (RIM et al., 2020): 4,556 blurry-sharp image pairs from 232 low-light static scenes. It contains both JPEG and RAW format. [DeepRFT]

**REDS** (NAH et al., 2019): 300 video clips (240 train, 30 validation, and 30 test) containing frames of size 1280×720. Proposed for both deblurring and super-resolution. [BasicVSR++; VRT]

**REDS4** (WANG et al., 2019): A subset of REDS containing 4 video clips, i.e., 000, 011, 015 and 020. Contains a 4-tuple comprising blur, blur\_bicubic (×4) and sharp\_bicubic (×4) degradations along with ground truth. [SRRUNet]

**Set14** (ZEYDE; ELAD; PROTTER, 2012): 14 classical computer vision images. Proposed for super-resolution.

**Set5** (BEVILACQUA et al., 2012): 5 images (baby, bird, butterfly, head, woman). Also proposed for super-resolution.

**UDM10** (YI et al., 2019): 10 video clips with frames at a resolution of 2K.

**Urban100** (HUANG; SINGH; AHUJA, 2015): 100 HR images collected from Flickr using keywords such as urban, city, architecture, and structure. [SRRUNet]

**Vid4** (LIU; SUN, 2014): 4 video clips. It contains two degradation types: Downsampling by bicubic interpolation (BIx4); and Gaussian blurring followed by downsampling (BDx4).

**Vimeo-90K** (XUE et al., 2019): 89,800 video clips downloaded from vimeo.com. It proposes 4 video restoration tasks: frame interpolation, denoising, deblocking, and super-resolution. [BasicVSR++; VRT]

**Waterloo Exploration Database** (MA et al., 2017): 4744 images. [DRUNet]

**In the Wild**: No specific dataset. Low-quality images are collected directly.

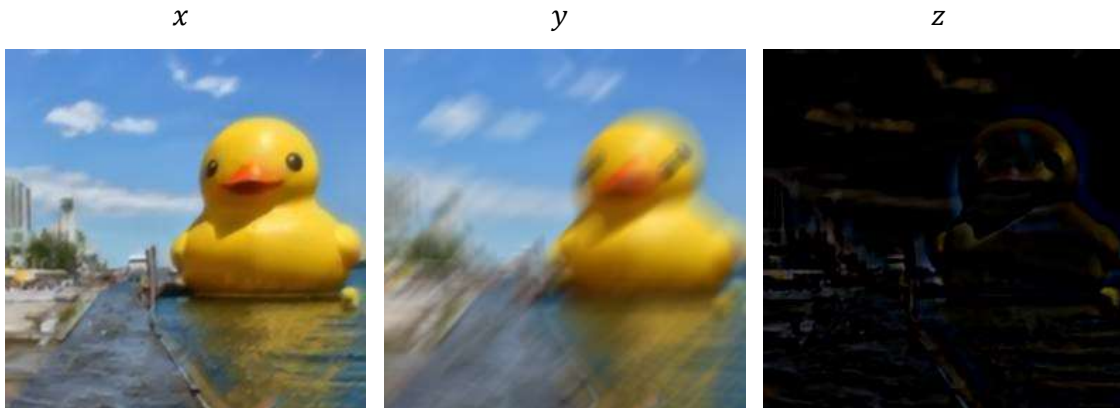
### 3 DEBLURRING

This chapter discusses both DeepRFT (MAO et al., 2021) which arms ResBlocks (NAH; KIM; LEE, 2017) with the Fourier Transform to better capture frequency discrepancies and Uformer (WANG et al., 2022) which combines a U-Net with Transformer blocks.

#### 3.1 DeepRFT

DeepRFT (**Deep Residual Fourier Transformation**) shoots for the sharp image directly without relying on kernel estimation. In order to achieve that, it models the degradation process as  $y = x - z$ , where  $y$  is the blurry image,  $x$  is the sharp image and  $z$  represents missing high-frequency information (i.e., edges and contours). Thus, the goal is to recover  $z$  and combine it with  $y$  leading to  $x = z + y$  which is shown in Figure 3.1.  $z$  might contain negative values which, when added to  $y$ , nullify undesired portions of the image while positive values enhance desired areas. To display  $z$  as shown below, these negative values are clamped to zero during save operation leading to mostly dark pixels.

Figure 3.1 – Degradation modeling where  $x = y + z$ .

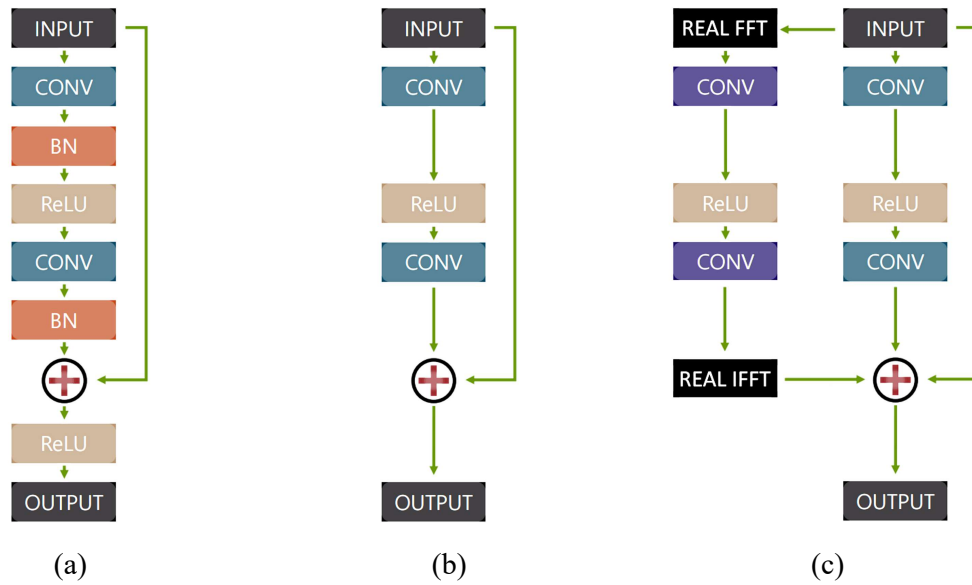


Source: the Author.

Since blurry and sharp image pairs are similar, it is more efficient to let the network learn the difference only, something that has been achieved with the ResBlock, a residual Conv-ReLU-Conv architecture. Residual learning (HE et al., 2016) correlates network depth with gains in accuracy. Surprisingly, with "plain" networks, adding nonlinear layers saturates

accuracy and, eventually, leads to higher error rates. Residual connections skip or shortcut one or more layers with a mapping that allows all information to be passed through (i.e., identity mapping). Compared to the original residual building block, the ResBlock benefits from faster convergence speed at training time, but its receptive field size is limited to stacking more ResBlocks which can largely increase computation. To efficiently account for global context, DeepRFT arms ResBlocks with a second stream based on a channel-wise FFT (i.e Res FFT-Conv Block) with  $1 \times 1$  convolutions. Figure 3.2 shows the three different building blocks.

Figure 3.2 – (a) Original residual network building block. (b) ResBlock. (c) Res FFT-Conv Block. CONV, BN and RELU stand for Convolutional, Batch Normalization, and ReLU layers, respectively.

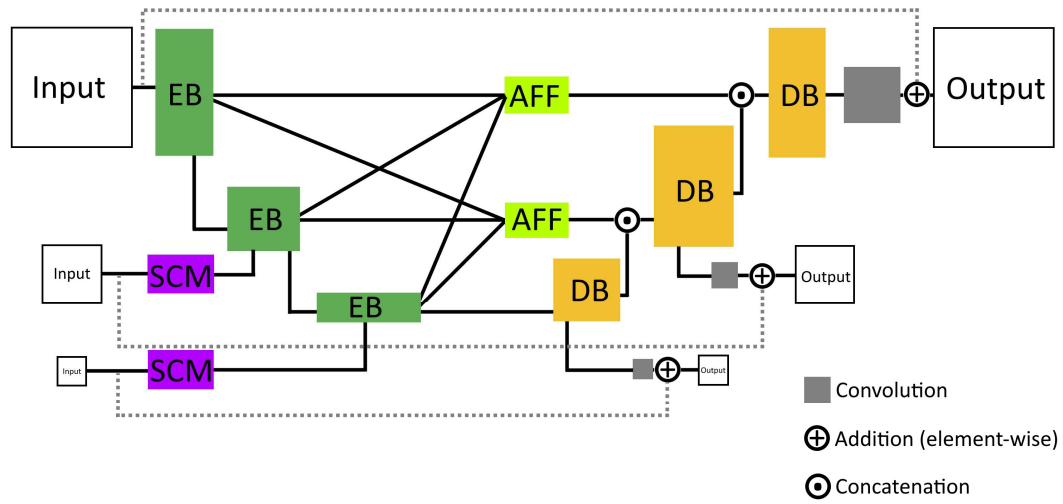


Source: adapted from Mao et al. (2021), Nah, Kim and Lee (2017).

DeepRFT itself is actually a combination of (i) Res FFT-Conv Blocks, (ii) the MIMO-UNet (CHO et al., 2021) backbone, and (iii) DO-Convs (CAO et al., 2022). MIMO-UNet handles different blur levels with multi-scale inputs, in particular, three downsampled versions of the input with scale factors 1, 2, and 4. While Figure 3.3 provides a first glance of MIMO-UNet, an in-depth look at the architecture can be found in the original work of Cho et al. (2021). DeepRFT has three variants: Small, Base, and Plus (+). DeepRFT+ is the most powerful one and is based on MIMO-UNet+ which encompasses 20 residual blocks for each Encoder Block (EB) and Decoder Block (DB). In contrast to MIMO-UNet, DeepRFT replaces all ResBlocks with Res FFT-Conv Blocks. Finally,  $3 \times 3$  convolutional layers are replaced by DO-Conv.



Figure 3.3 – MIMO-UNet backbone. Shallow Convolutional Module (SCM) extracts features and Asymmetric Feature Fusion (AFF) merges multi-scale features.



Source: the Author.

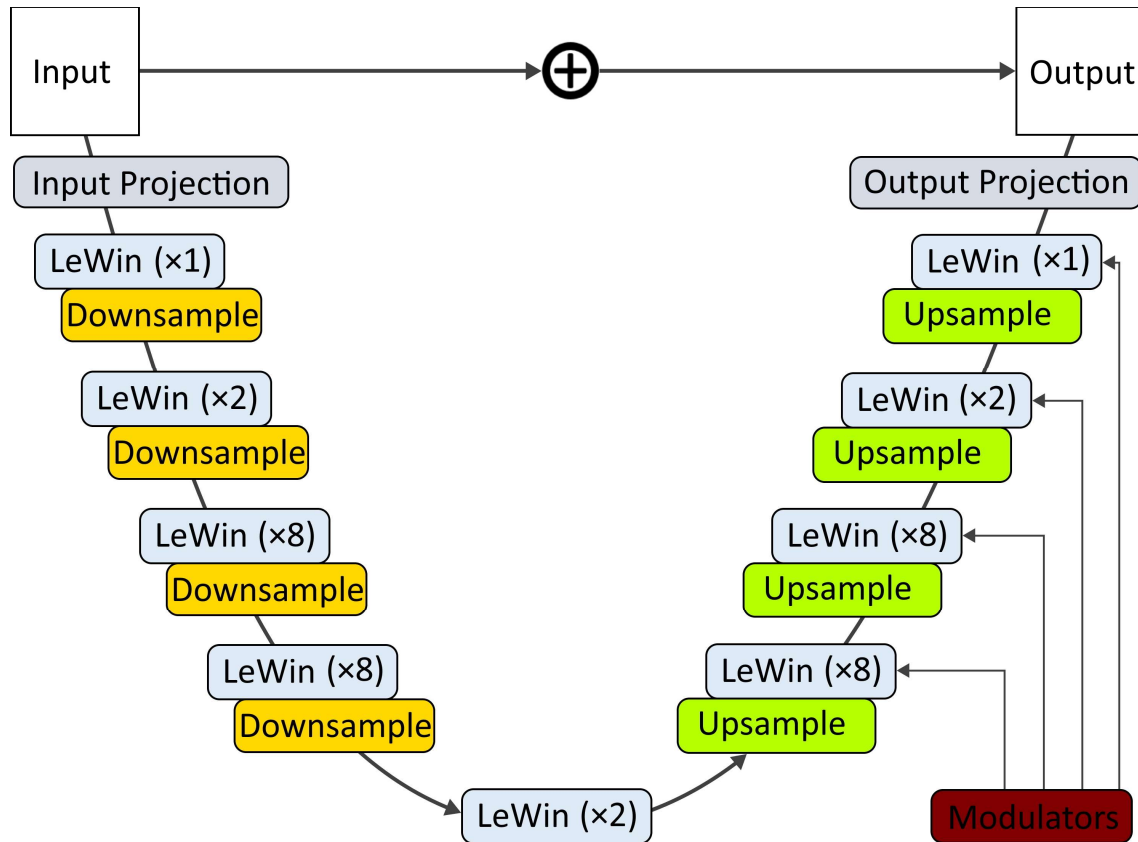
DO-Conv is a combination of conventional convolution (CC) and depthwise convolution (DC) that accelerates network training and slightly improves accuracy (e.g., PSNR). In a CC layer, the weight is composed of  $C_{out}$  kernels with each kernel having  $C_{in}$  channels, where  $C_{out}$  and  $C_{in}$  are the numbers of output and input channels, respectively. The output patches of a CC layer are given by the dot-products of the different (kernel, patch) pairs. Thus, the input channels end up getting mixed. In a DC layer, input channels are processed separately as the output patches are given by the dot-products of the different (kernel, patch) pairs per channel. The weight of a DC layer is made of  $D_{mul}$  kernels with each kernel having  $C_{in}$  channels, where  $D_{mul}$  is the depth multiplier which controls spatial dimensions of output patches. Actually, a DC layer can be implemented as a CC layer by employing grouped convolutions with  $C_{out} = C_{in} = \text{number of groups}$ . Finally, the DO-Conv combines both layers to mimic the functionality of the CC layer, but the overparameterization (ARORA; COHEN; HAZAN, 2018) yields the aforementioned benefits.

### 3.2 Uformer

Uformer (U-Shaped Transformer) also has three variants, i.e., Tiny (T), Small (S), and Base (B), with the latter being the most powerful. Figure 3.4 shows Uformer-B's structure.

Furthermore, Uformer also follows the approach of learning the difference between LQ-HQ image pairs, but it relies on Locally-enhanced Window (LeWin) transformer blocks. Finally, its design allows it to deal with tasks beyond deblurring such as denoising and deraining.

Figure 3.4 – Overview of Uformer-B. Both Input and Output Projections are 3×3 convolutional layers while the former adds a LeakyReLU activation.

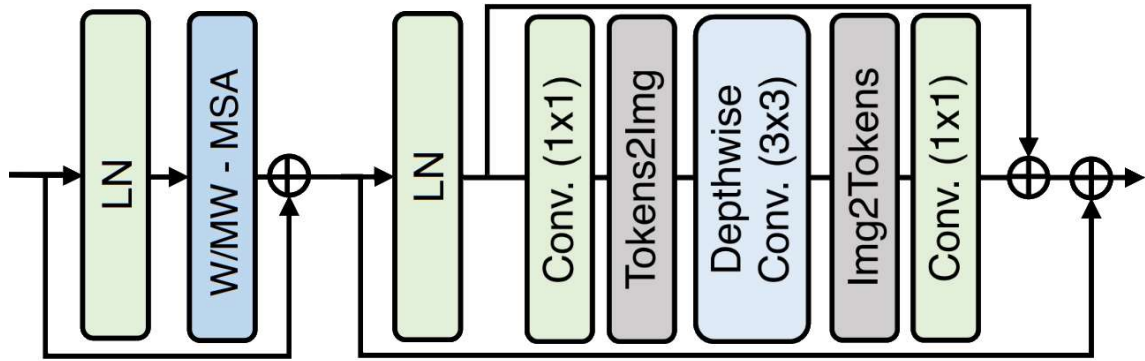


Source: the Author.

The original Transformer (VASWANI et al., 2017) is a self-attention based natural language processing (NLP) network which surpassed CNNs and RNNs in translation tasks, e.g., English-to-German, since it could be trained faster and achieved better results. Henceforth, transformer based-models such as BERT (DEVLIN et al., 2018) achieved a new state-of-the-art in NLP tasks. Later, Vision Transformer (ViT) (DOSOVITSKIY et al., 2020) successfully applied self-attention to images surpassing CNNs again with the condition that the training dataset was large enough, leading to interesting ramifications (RADFORD et al., 2021; PATASHNIK et al., 2021). However, ViT computes self-attention globally and is less capable of processing individual image patches. While the former is computationally prohibitive on

high-resolution images, the latter hinders the use of pixel neighborhoods which are useful in image restoration. The LeWin block is designed to account for those issues since (i) it computes self-attention within non-overlapping windows, thus reducing computational complexity, and (ii) it contains a  $3 \times 3$  depthwise convolutional layer, therefore leveraging local information.

Figure 3.5 – LeWin Transformer block. LN is Layer Normalization and W-MSA is Window-based Multi-head Self-Attention.



Source: adapted from Wang et al. (2022).

Figure 3.5 illustrates the LeWin block. Uformer computes self-attention once for every LeWin block, e.g., a total of forty times in Uformer-B. In fact, it is computed in W-MSA. Self-attention relates different positions of a sequence by mapping to an output, queries, keys, and values which are packed into three matrices:  $Q$ ,  $K$ , and  $V$ . Each  $Q$  matrix and each  $K V$  pair are obtained by passing the input signal which is composed by a set of non-overlapping windows through different single layer feed forward networks. In addition, a bias term  $B$  is added to all non-overlapping windows in order to calibrate features according to the degradation type, e.g., blur, noise, rain. Then, self-attention is computed as

$$\text{SoftMax}(B + QK^T / \sqrt{d_k})V. \quad (3.1)$$

Finally, *Tensor2Img* and *Img2Tokens* are tensor rearrangement operations (ROGOZHNIKOV, 2022) which let depthwise convolution operate on actual feature maps.

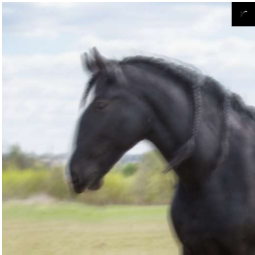



### 3.3 Uformer vs. DeepRFT

In sections such as this one, visual results are shown along with metrics and a brief discussion for illustrative purposes only. For a through qualitative and quantitative evaluation

with specific datasets, readers are encouraged to refer to the original papers. Experiments have been conducted with the original source code provided by the authors and, unless when explicitly stated, no modifications have been made.

In this section, experiments compare DeepRFT+ with Uformer-B. Figure 3.6 is a synthetic example which allows computing metrics due to the availability of a ground truth, i.e., clean image. In addition, the blur kernel is shown in the upper right-hand corner of the input picture. Although both methods achieved similar results, PSNR is quite low since it is a pixel-wise metric and, for this example, the synthetic blur kernel is off-centered which slightly shifts its convolution output, i.e., the blurry image. Also, both PSNR and SSIM do not correlate with human visual perception to image quality (YANG et al., 2021). Finally, LPIPS may be more reliable as it operates on extracted features.

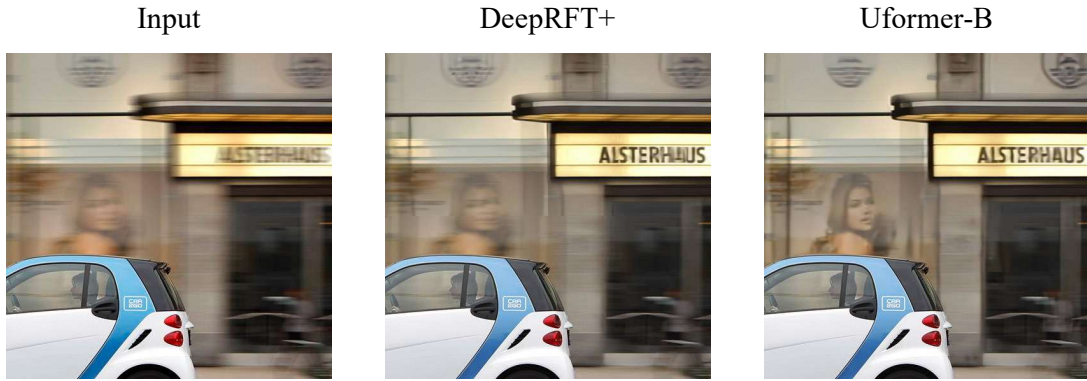
Figure 3.6 – Comparison of DeepRFT+ and Uformer-B applied to synthetic uniform blur removal.

Input	DeepRFT+	Uformer-B	Ground Truth
			
PSNR ↑	23.22	<b>23.35</b>	
SSIM ↑	0.733	<b>0.734</b>	
LPIPS ↓	<b>0.322</b>	0.334	

Source: the Author.

Figure 3.7 shows results which can only be assessed through visual inspection, i.e., qualitatively, because there is no ground truth image. Furthermore, it is a case of non-uniform blur which is not suited for kernel-based deblurring. Both methods achieve impressive results considering no kernel information is provided. However, even though Uformer has recovered more details, neither of them has recovered the lower right-hand corner of the image which appears to be part of a bicycle. Clearly, there are limitations to both DeepRFT+ and Uformer-B.

Figure 3.7 – Comparison of DeepRFT+ and Uformer-B applied to in the wild non-uniform blur removal.



Source: the Author.

On the whole, DeepRFT combines ResNet with a Multiple-Input Multiple-Output architecture, i.e., MIMO, whereas Uformer incorporates Transformers adapted for image restoration tasks into a U-shaped network. Moreover, Res FFT-Conv blocks are elegantly designed to account for global context using Fast Fourier Transform while depthwise convolution brings locality to LeWin blocks. Overall, both methods perform blind deblurring albeit with limitations. Finally, as already mentioned, Uformer also deals with other types of degradation which could mean squandered potential. If modulators were removed and Uformer was trained for deblurring only, better results might be achieved. Perhaps, the day will come when deep deblurring methods will be able to fully restore photos such as those from Street View.

## 4 SUPER-RESOLUTION

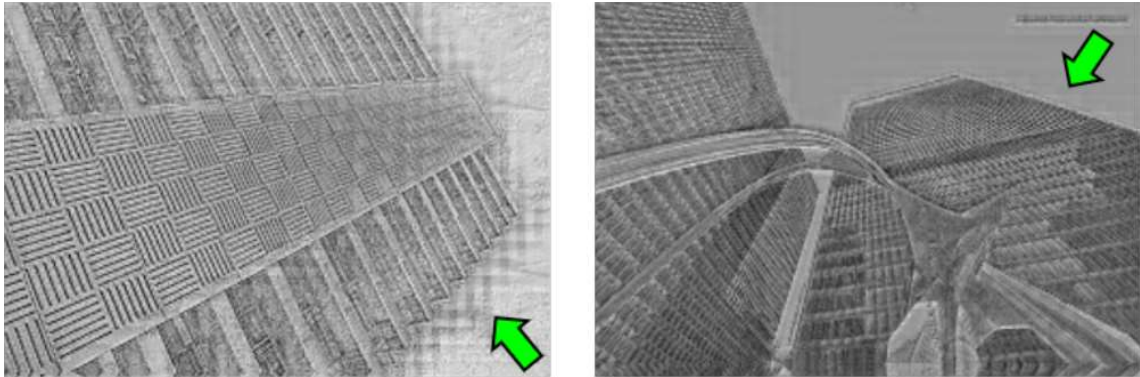
Similar to Chapter 3 which deals with single image deblurring, this chapter focuses on single image super-resolution (SISR). Deep learning applied to SISR can be traced back to the pioneering work of Dong et al. (2014) which proposed SRCNN: a sequence of three convolutional layers which mapped a low-resolution (LR) image to a high-resolution (HR) image. Since then, a surge of deep SR methods have eclouded (ANWAR; KHAN; BARNES, 2020). Notably, HAT (CHEN et al., 2022) and SROOE (PARK; MOON; CHO, 2022) can be considered representative SR methods because they are a perpetuation of previous successful SR architectures: SwinIR (LIANG et al., 2021) and ESRGAN (WANG et al., 2018b), respectively.

### 4.1 HAT

Upon contemplating strengths and weaknesses of SwinIR which comes from **Shifted Window** (Swin) Transformer (LIU et al., 2021), Chen et al. (2022) developed HAT (**Hybrid Attention Transformer**) which is a new Transformer architecture combining channel attention and self-attention to utilize more input information. Channel attention applies learnable weights to each feature channel, thus determining the importance of different channels according to a given task (YANG, 2020), e.g, super-resolution.

Swin Transformer introduced a shifted windowing scheme which limited self-attention to non-overlapping local windows, thus bringing greater efficiency. Inspired by that, SwinIR transferred the shifted-window strategy to image restoration and, notably, so did Uformer (Section 3.2). However, Chen et al. (2022) observed SwinIR produced feature maps with blocking artifacts (e.g., Figure 4.1) which were caused by the window partition mechanism of Swin Transformer, suggesting that the shifted window mechanism was inefficient to build the cross-window connection. Thus, to better aggregate the cross-window information, HAT employed an **Overlapping Cross-Attention Block** (OCAB) which enlarged the receptive field of window-based self-attention.

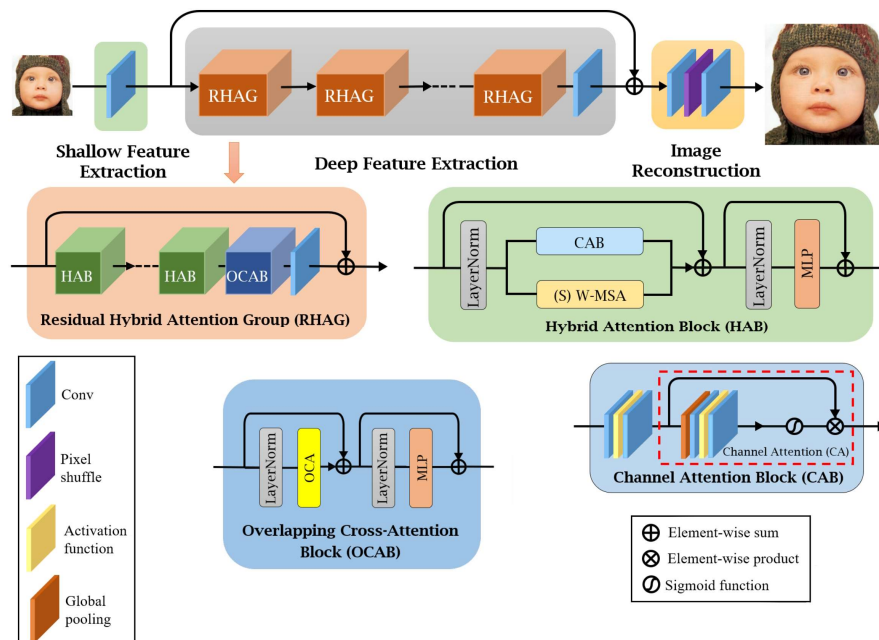
Figure 4.1 – Example of blocking artifacts in two different feature maps of SwinIR.



Source: Chen et al. (2022).

Figure 4.2 provides a glimpse at the architecture of HAT. Shallow Feature Extraction corresponds to an early  $3 \times 3$  convolutional layer (with  $C_{in} = 3$  and  $C_{out} = 180$ ) which improves visual representation, i.e., it helps upcoming transformer blocks see better leading to stable optimization. Deep Feature Extraction is composed of multiple RHAG blocks. Each block contains a residual connection which also stabilizes optimization, and one  $3 \times 3$  convolutional layer at the end (with  $C_{in} = C_{out} = 180$ ) which further combines the resulting deep features. Finally, Image Reconstruction upsamples the result of fusing shallow features with deep features by using  $3 \times 3$  convolutional layers intermingled with pixel-shuffle (SHI et al., 2016) which is a sub-pixel convolutional layer designed for super-resolution.

Figure 4.2 – HAT architecture.



Source: Chen et al. (2022).

## 4.2 SROOE

SROOE (Super-Resolution **Optimal Objective Estimation**) builds on ESRGAN (WANG et al., 2018b) which employs a generative adversarial network (GAN) (GOODFELLOW et al., 2014). GANs are trained under the minimax zero-sum game in which two players participate in an adversarial manner. Specifically, a model  $G$  learns to generate fake imagery which looks real enough to deceive a discriminator model  $D$ . Meanwhile,  $D$  learns to distinguish between real and fake samples. When adversarial training is complete,  $G$  is able to generate photo-realistic imagery. ESRGAN applies GAN technology to image super-resolution and SROOE improves ESRGAN by (i) incorporating spatial feature transform layers (WANG et al., 2018a) and (ii) first passing the LR input through a set of network layers, i.e., a predictive model which infers a region based objective map, i.e., an **Optimal Objective Estimation** (OOE) map. An objective can be defined as a desired outcome which can be further expressed by the result of minimizing a loss function. Regarding SROOE, such loss function is

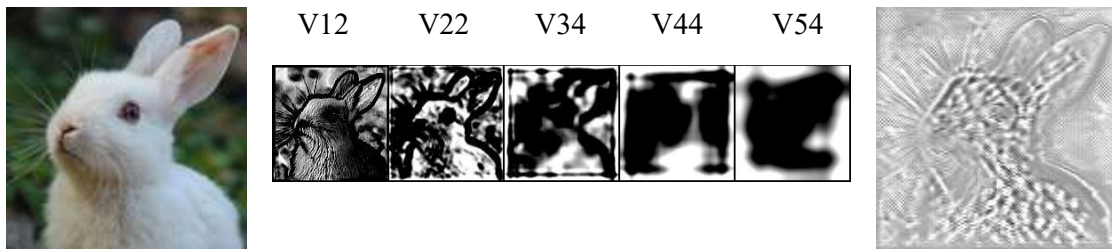
$$\mathcal{L} = (\lambda_{rec}\mathcal{L}_{rec}) + (\lambda_{adv}\mathcal{L}_{adv}) + \sum_{per_l}(\lambda_{per_l}\mathcal{L}_{per_l}) \quad (4.1)$$

which is composed of a reconstruction loss, e.g., L1, an adversarial loss, and a perceptual (VGG) loss with  $per_l \in \{V12, V22, V34, V44, V54\}$ , a set of VGG features. However, it is challenging to find optimal values for each lambda term (or regularizer). Besides, different regions of an image have different optimal values for these lambda terms, e.g., the nose region of a selfie photo has more high-frequency details than the forehead region and, thus, these regions ought to be treated differently. Therefore, the goal of using an OOE map is to find the optimal loss objective for each image pixel.

OOE maps are inferred by a predictive model which is a combination of VGG and a U-Net. VGG features  $V12$ ,  $V22$ ,  $V34$ ,  $V44$ , and  $V54$  are extracted from the LR input, concatenated and then passed to a U-Net. The U-Net is composed of eighteen  $3 \times 3$  conv layers interspersed with batch normalizations and ReLUs, and one final  $1 \times 1$  conv layer. As illustrated in Figure 4.3, the output OOE map is a single channel image which is then passed to ESRGAN along with the LR input.



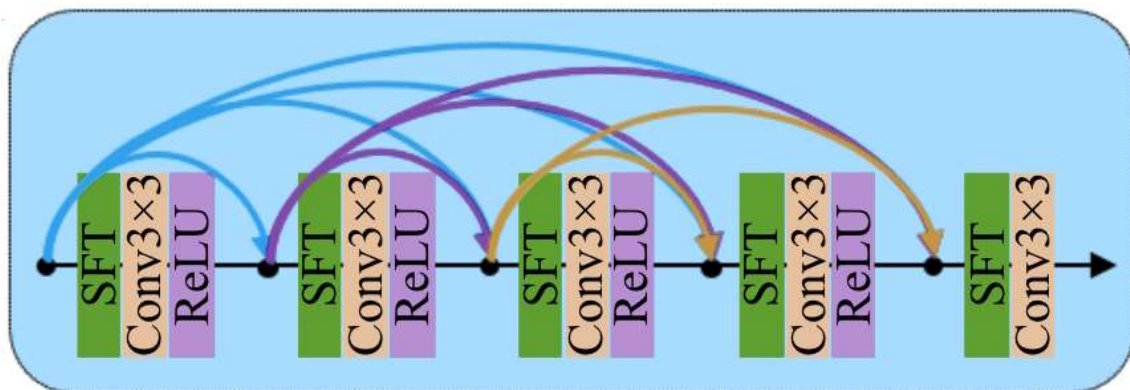
Figure 4.3 – LR input on the left side, some of its (normalized) VGG features in the middle, and corresponding OOE map on the right side.



Source: the Author.

SROOE's ESRGAN model is composed of 23 **Residual-in-Residual Dense Blocks** (RRDB) also referred to as Basic Blocks which, similar to ResBlocks, are devoid of batch normalization while still employing residual connections. Each RRDB is composed of 3 Dense Blocks and each Dense Block contains 4 Spatial Feature Transforms (SFTs), as illustrated by Figure 4.4. SFT learns a mapping function that outputs a modulation parameter allowing ESRGAN to optimize the changing objective during training and to generate SR results with spatially different objectives according to the map at inference time. Simply put, SFT allows changing the network behavior according to the objective map.

Figure 4.4 – A single Dense Block.








Source: the Author.

### 4.3 SROOE vs. HAT

In the first experiment illustrated in Figure 4.5, the LR image is the bicubically downsampled version of the ground truth with scale factor  $sf = 4$ . Interestingly, SROOE performs better perceptually (LPIPS) while HAT scores higher PSNR and SSIM.

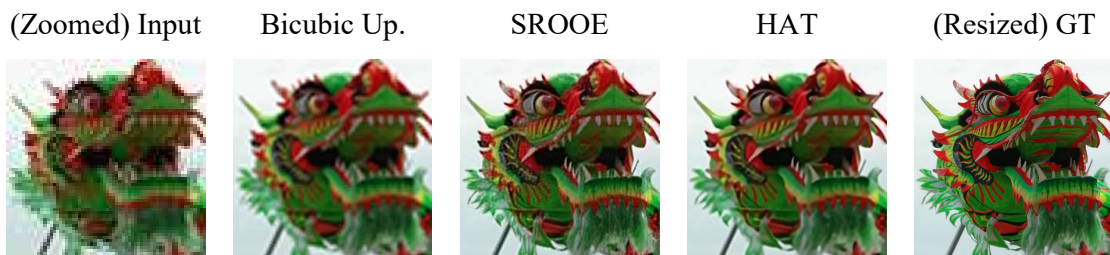
Figure 4.5 – Super-resolution comparison with scale factor  $sf = 4$ .

(Zoomed) Input	Bicubic Up.	SROOE	HAT	Ground Truth
				
PSNR ↑	30.37	30.77	<b>31.84</b>	
SSIM ↑	0.819	0.822	<b>0.861</b>	
LPIPS ↓	0.318	<b>0.160</b>	0.257	

Source: the Author.

Although the released models of both SROOE and HAT only support using scale factors of at most four, it is sometimes desirable to go beyond that. In fact, real LR images could have icon size, e.g.,  $64 \times 64$ , which implies the use of larger scale factors. In order to see how SROOE and HAT would deal with that, Figure 4.6 shows a LR input image which has been bicubically downsampled using  $sf = 16$  along with the corresponding  $1024 \times 1024$  ground truth image. While SROOE and HAT have generated better results when compared to simple bicubic upsampling, their output size for this example is  $256 \times 256$  due to the aforementioned limitation.

Figure 4.6 – Super-resolution comparison with scale factor  $sf = 16$ . Metrics cannot be computed for images of different sizes the reason why they are not shown.



Source: the Author.

To sum up, SROOE is a perception-oriented model which yields better LPIPS and employs VGG features, an objective estimator, and a generator. Meanwhile, HAT seeks to activate more pixels through combined channel attention and self-attention leading to better PSNR and SSIM. Actually, even though PSNR is the most widely used metric in image restoration, it is also related to the MSE loss which favors over-smoothed predictions (ZHANG et al., 2022) and, therefore, does not reflect actual human visual response. Regardless of metrics, the current challenge of deep SISR methods is to scale to larger scale factors which, so far, only successfully go up to  $\times 4$ .

## 5 NON-BLIND RESTORATION

The two methods discussed in this chapter simultaneously deal with deconvolution, upsampling and denoising, and share a single degradation model:

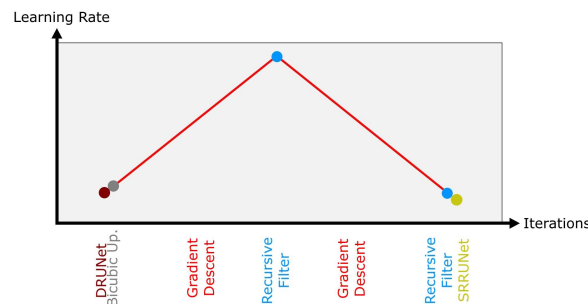
$$y = (x \otimes k) \downarrow_s + n, \quad (5.1)$$

where  $\otimes k$  is the convolution operation with blur kernel  $k$ ,  $\downarrow_s$  is the downsampling operation with scale factor  $s$ , and  $n$  is additive noise with noise level  $\sigma$ . Moreover, the concept "non-blind" states that these methods require additional information to be provided as inputs beyond the low-quality image  $y$ , i.e., blur kernel  $k$ , scale factor  $s$  and noise level  $\sigma$ . This kind of inverse problem often has a data term which is meant to obtain the image  $x$  that corresponds to the given degradation model, and a prior term which imposes natural image characteristics on  $x$ . Such data and prior term paradigm is accurately followed by USRNet (ZHANG; GOOL; TIMOFTE, 2020) which leverages frequency domain properties. In contrast, Our Non-Blind is more concerned with undoing each degradation operation by backtracking the degradation model. Finally, both methods use deep denoisers.

### 5.1 Our Non-Blind

Given Equation 5.1, it is intuitive to first deal with removing noise, then upsampling, and lastly deconvolution. Additive noise is handled by the pretrained model of DRUNet (ZHANG et al., 2021). Next, bicubic upsampling is used. We devise a method called GDDec where Gradient Descent is used directly on the image along with the smoothing filter of Gastal and Oliveira (2011) in order to suppress deconvolution artifacts. The last step is improving image resolution which is achieved by SRRUNet. Figure 5.1 provides a glimpse of Our Non-Blind.

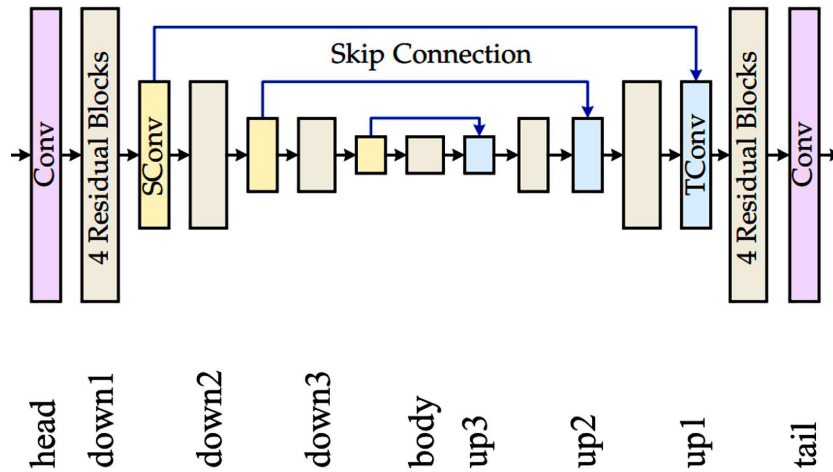
Figure 5.1 – Main components of Our Non-Blind.



Source: the Author

DRUNet (**D**enoising **R**esidual **U**-**N**et) combines a U-Net backbone with 28 ResBlocks. While Chapter 3 has already described residual learning in detail, U-Nets are further discussed in Chapter 7. DRUNet takes as input a noisy image concatenated with its corresponding noise level map. This noise level map is simply the noise level  $\sigma$  repeated along the dimensions of the image, meaning this map becomes an extra channel (e.g., a fourth channel in the case of RGB images). This allows for a simple and efficient design as the noise level is never dealt with explicitly and, furthermore, it eventually disappears within the hidden nodes of the network. There are four scales to DRUNet: 64, 128, 256, and 512 channels. In U-Net architectures, the larger the scale, the smaller the feature map is in terms of width and height. DRUNet's blueprint can be visualized in Figure 5.2.

Figure 5.2 – DRUNet's architecture.



Source: adapted from Zhang et al. (2020).

### 5.1.1 GDDec

As of now, gradient descent is the predominant way of training neural networks. In addition, present day methods employ Adam (KINGMA; BA, 2014) which is a gradient descent scheme based on momentum and preconditioning that stabilizes and accelerates optimization. However, optimizing weights and biases during training is not as direct as optimizing the input image itself during evaluation. In some situations, one may skip the time-consuming task of training a network for a specific end, while preserving efficacy. Accordingly, GDDec is our proposed **G**radient **D**escent **D**econvolution approach which is applied to deblurring. GDDec removes blur in an iterative process by minimizing

$$\mathcal{L} = 1 - SSIM((y' \otimes k) \downarrow_s, y), \quad (5.2)$$

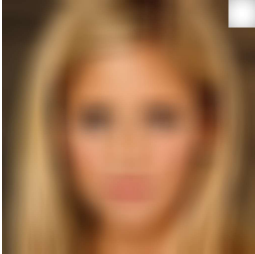
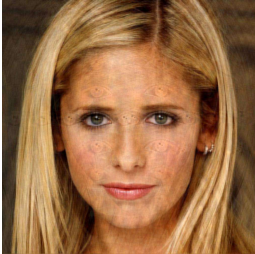
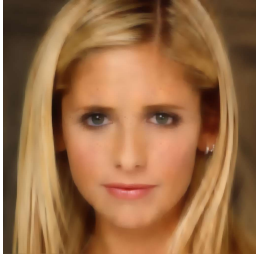
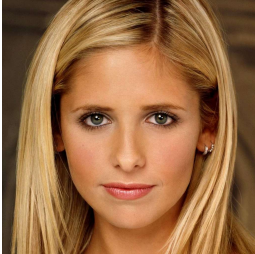
where  $y$  is the low-quality input image and  $y'$  is the gradient descent parameter which, upon the end of optimization, approximates to the desired high-quality image  $x$ , e.g.,  $y' \approx x$ . Besides being a metric, SSIM can also be used as a loss function (ZHAO et al., 2017). In most of our experiments, it has led to better results than when using L1 or L2 losses. The parameter  $y'$  is initialized as an upsampled version of  $y$  using bicubic interpolator with the given scale factor. Although that does not imply super-resolution, it allows handling the complete degradation model mentioned in the beginning of this chapter.

Similar approaches have been tried in the past with RGDN (GONG et al., 2018) which incorporates neural networks into a fully parameterized gradient descent scheme to iteratively perform deconvolution, and Deep Image Prior (ULYANOV; VEDALDI; LEMPITSKY, 2017) which employs neural networks with randomly initialized weights and zero training. Different from RDGN which learns a specific optimizer, GDDec uses the off-the-shelf Adam optimizer which requires zero training since it has no learnable parameters. Also, unlike Deep Image Prior, no neural networks are used in GDDec. Inspired by the implementation of Menon et al. (2020), a learning rate schedule  $f(t, T)$  is used, where  $t$  denotes the current iteration and  $T$  is the maximum number of iterations. Specifically, GDDec adopts *linear1cycle* which is defined as

$$f(t, T) = \left(1 + 9 \left(1 - 2 \left\lfloor \frac{t}{T} - \frac{1}{2} \right\rfloor\right)\right) / 10. \quad (5.3)$$

It allows a natural warm-up, peak activity, cooldown process which empirically gives the best results. Learning rate schedules work as follows: at any given time  $t$ , the current learning rate of the optimizer is given by the initial learning rate value  $LR$  multiplied by  $f(t, T)$ , where  $LR = 0.05$  in all experiments, and  $T$  varies according to the amount of blur. Also, for faster computation, Equation 2.2 can be used, i.e., the convolution theorem. Nonetheless, as shown in Figure 5.3, deconvolution of severely blurred images is subject to distracting artifacts such as ringing. In order to deal with that, Our Non-Blind relies on a real-time edge-aware smoothing filter.

Figure 5.3 – Results of GDDec and results of removing ringing artifacts using the recursive filter (RF) with  $(\sigma_s, \sigma_r) = (60, 0.5)$ .

Input	GDDec	GDDec with RF	Ground Truth
			
PSNR ↑	<b>29.22</b>	26.43	
SSIM ↑	<b>0.807</b>	0.716	
LPIPS ↓	<b>0.359</b>	0.465	

Source: the Author.

The recursive filter of Gastal and Oliveira (2011) preserves strong edges and has been used in the past (FORTUNATO; OLIVEIRA, 2014) to remove deconvolution artifacts. Notably, as illustrated by Figure 5.3 the recursive filter can be used to eliminate ghosts associated with ringing. It contains parameters  $(\sigma_s, \sigma_r)$  which are inherited by Our Non-Blind and its number of iterations is fixed at 3. It is interesting, however, that metrics may not always be indicators of image quality.

It is worth noting that available blur kernel and actual blur kernel may differ due to kernel estimation imprecisions which generally correlate with kernel size. Of course, this does have an effect on deconvolution algorithms and hampers the quality of result. In order to see how blur kernel imprecisions affect GDDec, we use the method of Pan et al. (2016) to estimate a blur kernel which is shown in Figure 5.4. Then, the estimated kernel is used by GDDec to deblur the corresponding leftmost image shown in Figure 5.5. Clearly, GDDec is sensitive to kernel imprecisions leading to more ringing. In such cases, it is useful to change the loss function to L1 loss which leads to less sharp images, but also leads to less ringing.

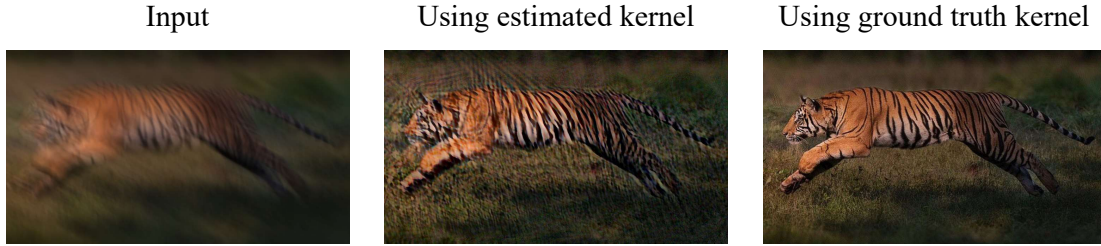
Figure 5.4 –  $69 \times 69$  Blur kernel estimated by the method of Pan et al. (2016).



Source: the Author.



Figure 5.5 – Results of GDDec when using an estimated blur kernel.



Source: the Author.

### 5.1.2 SRRUNet

The final step is to super-resolve the output image which may still be at a lower-resolution despite having been previously resized. To that end, a second DRUNet model is fine-tuned, i.e., retrained, for the specific task of super-resolution, thus arriving at SRRUNet (Super-Resolution Residual U-Net). Similar to DRUNet in which a noise level map is constructed from the noise level, a scale factor level map (or degradation map) can be obtained by repeating the scale factor level along the image height and width. The scale factor level is defined as

$$(sf - 1)/sf_{max} \quad (5.4)$$

where  $sf$  is any value from  $[1, 16]$  and  $sf_{max}$  is set to 16. Thus, this degradation map is concatenated to the image and the network implicitly deals with it in order to restore the image.

The fine-tuning dataset is composed of 8,856 HR images collected from seven datasets, i.e., AFHQ, FFHQ, Flickr1024, GOPRO, REDS4, Manga109, and Urban100. Such diverse dataset includes various images of animals, people, vegetation, buildings, cars, cartoons, complex architectures and more. During fine-tuning, patches of size  $496 \times 496$  are used and the dataset is further augmented with random horizontal flips. Moreover, a batch size of 1 is used.

LR-HR image pairs are synthesized using the following degradation model which preserves image size:

$$y = x \downarrow_{s1} \uparrow_{s2} \quad (5.5)$$

where the  $\downarrow/\uparrow$  arrow indicates a resizing operation using scale factors  $s1$  and  $s2$ . The value of  $s1$  is an integer randomly and uniformly chosen from  $[1, 16]$ , whereas  $s2$  is calculated at runtime to restore the original height and width of the image. Since Our Non-Blind has already


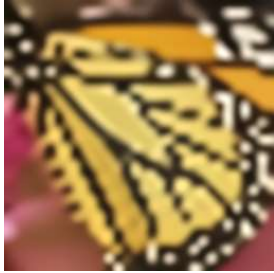




resized the image, it stands to reason that SRRUNet ought to be a size preserving network which differs from the SR methods discussed in Chapter 4. Put it simple, SRRUNet is concerned with increasing resolution without affecting image size. The motivation for using larger scale factors comes from Yang et al. (2021) which successfully trains a network to deal with scale factors up to 200, albeit in a domain dependent manner. The downsampling algorithm used is bicubic and the upsampling algorithm is randomly chosen from [bicubic, nearest neighbor], thus imbuing some degree of robustness to the network, since bicubic and nearest produce visually different results.

Fine-tuning lasts for 128 epochs. Learning rate schedule is adapted from Zhang et al. (2021), i.e., learning rate starts at  $2e-4$  and decreases by 5% every epoch. The loss function is  $\mathcal{L} = \lambda_{rec} \|HR - GT\|_1 + \lambda_{per12} \|V12(HR) - V12(GT)\|_1 + \lambda_{per22} \|V22(HR) - V22(GT)\|_1$ , where  $HR$  and  $GT$  denote SRRUNet's output and the ground truth respectively,  $\| \cdot \|_1$  is the L1 norm, and  $V12(\cdot)$  and  $V22(\cdot)$  compute different VGG19 features. Inspired by SROOE (PARK; MOON; CHO, 2022), the values of the regularizers are:  $\lambda_{rec} = 0.01$ ,  $\lambda_{per12} = 0.5$ ,  $\lambda_{per22} = 0.5$ . For model updating, Adam algorithm is used once more. Finally, the model is trained on an NVIDIA GeForce GTX 1070 GPU.

Figure 5.6 shows results before and after fine-tuning using an image from Set5. While PSNR and SSIM have lowered due to greater pixel distortion after fine-tuning, LPIPS has improved which better correlates with perceptual quality. While it would be interesting to compare the results of SRRUNet with those of SROOE and HAT, it would also be an unfair comparison since SRRUNet is trained to deal with various scale factors in a non-blind manner, whereas SROOE and HAT focus exclusively on a specific scale factor.

Figure 5.6 – Before and after fine-tuning SRRUNet.

(Zoomed x8) Input	Before	After	Ground Truth
			
PSNR ↑	<b>15.61</b>	15.09	
SSIM ↑	<b>0.460</b>	0.429	
LPIPS ↓	0.543	<b>0.409</b>	

Source: the Author

## 5.2 USRNet

USRNet (Unfolding Super-Resolution Network) rigorously separates data term and prior term. The data term is simply the solution of an equation (ZHAO et al., 2016) which leverages the Fast Fourier Transform. Meanwhile, a deep denoising/smoothing network, called ResUNet, functions as a prior. Like DRUNet, it also takes as input an image concatenated with a noise level map and outputs a cleaner image. In fact, USRNet, its ResUNet, and DRUNet are all creations of roughly the same authors. The only architectural difference is that ResUNet has half the number of ResBlocks. Furthermore, ResUNet has been trained to work in conjunction with the whole USRNet as opposed to DRUNet which has been trained in isolation. The decoupling between data and prior terms is better depicted in Algorithm 1 which provides a glimpse at how USRNet combines its own modules.

---

**Algorithm 1:** The core of USRNet.

---

```

1   $y, k, s, \sigma \leftarrow$  (degraded image, blur kernel, scale factor, noise level)
2   $h \leftarrow$  hypa( $\sigma, s$ )
3  for  $i \leftarrow 1 \dots 8$  do
4       $y \leftarrow$  data( $y, k, s, h$ )
5       $y \leftarrow$  prior( $y, h$ )
6  end for
7  return  $y$ 

```

---


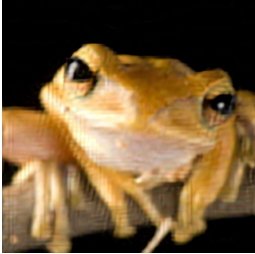
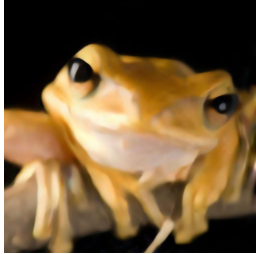

USRNet also has a hypa module, i.e., a deep hyperparameter estimator which takes as input the concatenation of noise level and scale factor, and yields hyperparameters  $h = [\alpha_1, \dots, \alpha_8, \beta_1, \dots, \beta_8]$  where the index corresponds to the current iteration. The module itself consists of three  $1 \times 1$  convolutional layers and three activations (two ReLUs and one Softplus) interleaved.

## 5.3 USRNet vs. Our Non-Blind

Figure 5.7 displays results of restoring a noise-free image with scale factor 16 and the blur kernel shown in the upper right-hand corner of the input. Clearly, PSNR and SSIM are not

accurate measures of perceptual quality. While USRNet's data term struggles to deal with scale factors above 4, Our Non-Blind has fully eliminated jagged lines yielding a lower LPIPS.

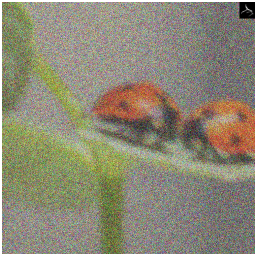

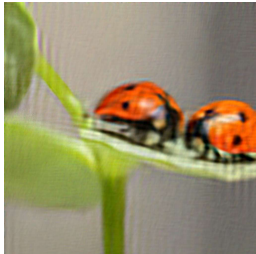

Figure 5.7 – Noise-free low resolution image.

Zoomed (x16) Input	USRNet	Our Non-Blind	Ground Truth
			
PSNR ↑	<b>28.78</b>	26.22	
SSIM ↑	<b>0.855</b>	0.781	
LPIPS ↓	0.367	<b>0.320</b>	

Source: the Author.

Figure 5.8 shows a different a scenario with an image corrupted by 60% gaussian noise, scale factor 1, and the blur kernel shown in the upper right-hand corner. Remarkably, both methods were able to reduce noise level significantly. This time, the metrics have favoured Our Non-Blind which inherits its denoising power from DRUNet.

Figure 5.8 – Extremely noisy image.

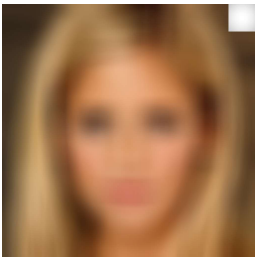
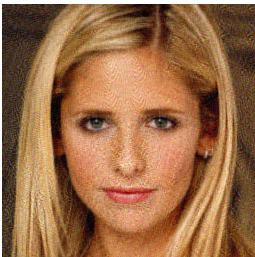
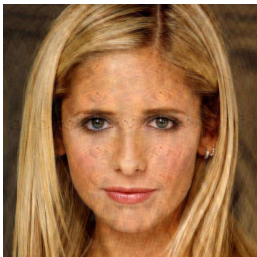
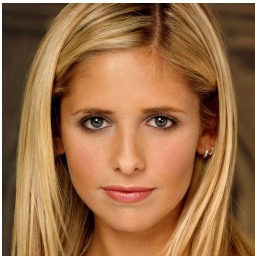
Input	USRNet	Our Non-Blind	Ground Truth
			
PSNR ↑	21.60	<b>24.22</b>	
SSIM ↑	0.521	<b>0.780</b>	
LPIPS ↓	0.705	<b>0.523</b>	

Source: the Author.

As an ablation study, Figure 5.9 illustrate the effects of using only USRNet's data module or only Our Non-Blind's gradient descent scheme. USRNet's hyperparameter generator





is still kept. Strikingly, Our Non-Blind's deconvolution output scores better in terms of metrics. The reason is deconvolution is sensitive to the mathematical imprecisions. USRNet's hypermodule, as well as neural networks in general, are actually approximations to desired functions or mappings (HORNİK, 1991). As a consequence, hyperparameters generated by the hypermodule have limited accuracy which is reflected by the outputs of USRNet's data term. Often, however, these approximations can be neglected since they still yield very compelling results. Lastly, Figure 5.10 illustrates DRUNet's superiority in Gaussian denoising.

Figure 5.9 – Gradient descent vs. USRNet's data module. Input is a noise-free image with large blur kernel and a scale factor of 1.

Input	USRNet's Data	Gradient Descent	Ground Truth
			
PSNR ↑	20.35	<b>29.22</b>	
SSIM ↑	0.205	<b>0.807</b>	
LPIPS ↓	0.693	<b>0.359</b>	

Source: the Author.

Figure 5.10 – DRUNet vs. ResUNet. Input is blur-free image with 50% gaussian noise and a scale factor of 1.

Input	ResUNet	DRUNet	Ground Truth
			
PSNR ↑	19.49	<b>24.26</b>	
SSIM ↑	0.498	<b>0.722</b>	
LPIPS ↓	0.424	<b>0.237</b>	

Source: the Author.

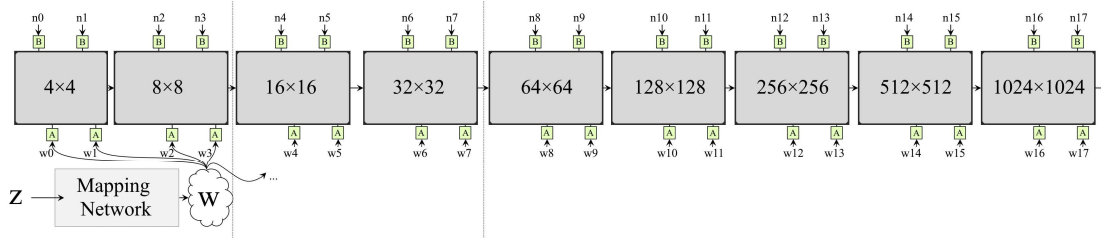
Since the GDDec part of Our Non-Blind is slow, it is difficult to report results for multiple images. Furthermore, USRNet's test script also has a bottleneck when the HQ image is first convolved with a blur kernel (to generate the LQ image) which is implemented using NumPy and, therefore, cannot be parallelized. Since Our Non-Blind is still an experimental method, we have only show results for some images, mainly for illustrative purposes which is in agreement with the rest of this thesis.

In conclusion, both methods share a single degradation model and employ neural networks. Our Non-Blind additionally uses a recursive filter as prior to handle ringing. As for the data term, USRNet's obtains the solution of a closed form expression while Our Non-Blind performs gradient descent on the image guided by the degradation model. Moreover, USRNet includes a hyperparameter generator which makes it very practical. Finally, combining the strengths of USRNet and Our Non-Blind could ultimately lead to a better model, but that is left for future work.

## 6 FACE RESTORATION

This chapter discusses the use of GAN (GOODFELLOW et al., 2014) technology in the field of Blind Face Restoration (BFR). Because GANs learn to generate images from a specified domain (e.g., faces, cars, bedrooms), they can also function as priors in blind image restoration, thus helping to constrain the problem. Since BFR aims to obtain realistic and high-quality images, both methods discussed below use the powerful StyleGAN (KARRAS; LAINE; AILA, 2018) image generator in some way, although neither benefits from the well known style mixing capabilities. GPEN (YANG et al., 2021) makes use of StyleGAN2's (KARRAS et al., 2020) architecture, while PULSE (MENON et al., 2020) leverages a pre-trained StyleGAN1 model. Figure 6.1 provides a glance at StyleGAN1's architecture. Noise inputs  $n_i$  and intermediate latent code  $w_i$  are broadcast across all resolution blocks. Feature maps flow out of each block, except for the last one which outputs a realistic high-quality  $1024 \times 1024$  image. Actually, both StyleGANs are very similar. They receive as inputs latent code  $z$  and noise  $n_i$  sampled from the Gaussian distribution and generate an image. The relevant difference here is that images are more easily projected into StyleGAN2's latent space.

Figure 6.1 – StyleGAN overview



Source: the Author.

### 6.1 GPEN

GPEN (GAN Prior Embedded Network) is a U-Net (RONNEBERGER; FISCHER; BROX, 2015) which learns a one-to-one mapping between low-quality face images and high-quality face images. It is a seamless combination of a CNN encoder and a GAN decoder that enjoys fast computation and often preserves face identity. As such, it is a suitable tool for face image restoration.

The official implementation of GPEN adopts StyleGAN2 as its GAN decoder. Because StyleGANs are not fully convolutional (i.e., they also employ an MLP for latent space mapping)

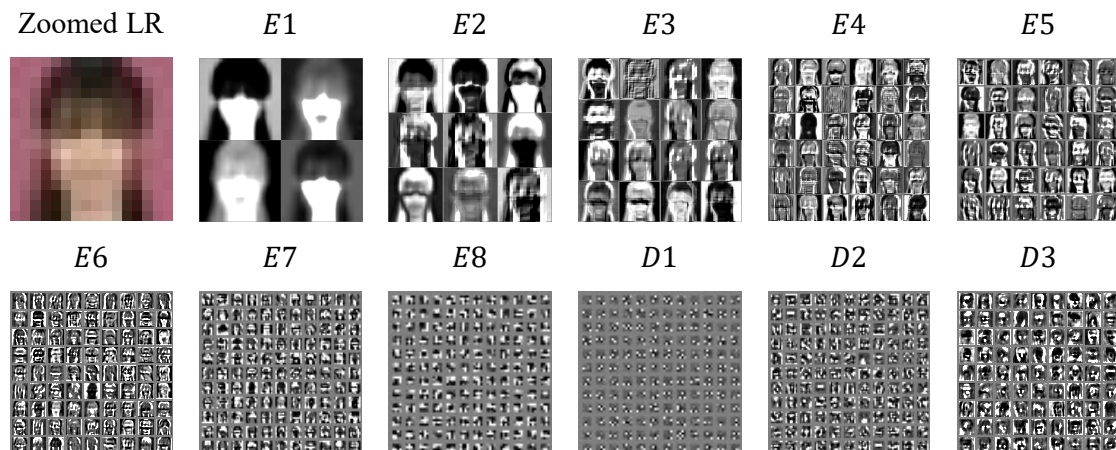


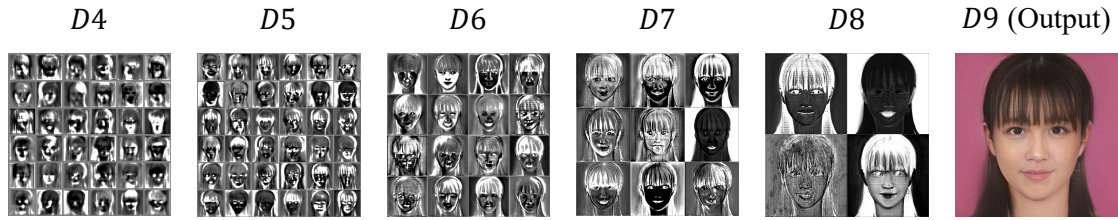
GPEN first resizes the input image to StyleGAN's resolution. Furthermore, all StyleGANs map Gaussian noise along with a latent code  $z$  (i.e., one that follows a Gaussian distribution) to a realistic image. Likewise, GPEN starts with a CNN encoder which maps the low-quality resized input image to DNN features. Then, latent code  $z$  and noise inputs to the trained StyleGAN2 network are replaced by deeper and shallower DNN features, respectively. However, merely combining the two architectures does not ensure that the DNN will generate suitable inputs to the GAN. To allow both encoder and decoder to adapt to each other, the entire U-Net is fine-tuned with a set of synthesized LQ-HQ face image pairs.

The CNN encoder is composed of eight convolutional layers each with a Fused Leaky ReLU activation (i.e., scaled LReLU). Apart from the first layer which employs  $1 \times 1$  convolutions, remaining layers use  $2 \times 2$  strided convolutions. Starting from a low-quality RGB image, the CNN encoder progressively shrinks the image while increasing its number of channels, eventually leading to features that can be used as latent code and noise. Then, the StyleGAN2 decoder progressively grows a small prototype multichannel image using latent code and noise obtained while decreasing the number of channels, eventually leading to a high-quality RGB image.

Figure 6.2 illustrates GPEN's U-Net behaviour with feature map channels generated during encoding (i.e., contracting) and decoding (i.e., expanding). Note that, unlike RGB images, these feature maps are composed of hundreds of channels and there are different methods of visualizing them. Here, they are displayed as a grid of individual channels which have been normalized by instance normalization (ULYANOV; VEDALDI; LEMPITSKY, 2016). In addition, only a few channels are shown per layer.

Figure 6.2 – Normalized feature maps for each layer of GPEN. The encoder fmap at layer  $k$  is denoted by  $E_k$ . Likewise for the decoder.





Source: the Author.

## 6.2 PULSE

PULSE (Photo Upsampling via Latent Space Exploration) is an optimization framework that employs a given pre-trained generative model, originally StyleGAN1. PULSE relaxes the face identity constraint, but ensures that the output is an HR face image. It does so using a clever design: the goal is to find the HR image that correctly downscales to the LR image. This downscaling increases ill-posedness since, as illustrated by Figure 6.3, different HR images can be downscaled to approximately the same  $16 \times 16$  image. However, since GANs are domain-based generators, the infinite solutions that may arise will always be of that domain (e.g., faces in StyleGAN1). So long as the chosen generative model generates high-quality, high-resolution imagery, so will PULSE. Thus, PULSE leverages ill-posedness to its own favour. Finally, downscaling a degraded image mitigates effects of blur and noise, which is what enables PULSE to operate as a blind method.

Figure 6.3 – Downscaling images leads to loss of information. When the resulting face image is very small (e.g.,  $16 \times 16$ ) there is no way of telling whose face it is.



Source: the Author.



In practice, PULSE achieves its goal by optimizing a latent code in  $Z$  latent space.  $Z$  space is used because it is Gaussian and, therefore, shaped as a sphere or bubble (as opposed to intermediate  $W$  space of StyleGAN, whose shape is not explicitly known). This property, when aligned with spherical optimization, ensures the optimizer stays on the natural image manifold (i.e., the one used to train the generative model), avoiding unrealistic face images. To actually find the desired latent code  $z$ , the optimizer minimizes

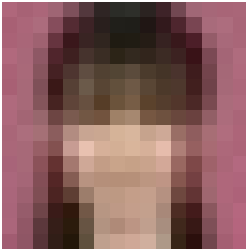



$$\|D(G(z)) - I\|, \quad (6.1)$$

where  $I$  is the LR input,  $G$  is the generator, and  $D$  is the downscaling function. Once that  $z$  code is found, the final output can be obtained from  $G(z)$ .

### 6.3 PULSE vs. GPEN

Figure 6.4 shows a synthetic example where the original (ground truth) photo was degraded using an  $91 \times 91$  gaussian blur kernel and a scale factor of 64. Although both PULSE and GPEN have generated high-quality realistic outputs, neither preserved identity. Despite that, GPEN better recovered ethnicity and hair color.

Figure 6.4 – Computer simulated degradation leading to a  $16 \times 16$  image.

Zoomed LR ( $\times 64$ )	PULSE	GPEN	Ground Truth
			
PSNR $\uparrow$	19.99	<b>22.42</b>	
SSIM $\uparrow$	0.676	<b>0.764</b>	
LPIPS $\downarrow$	0.484	<b>0.387</b>	

Source: the Author.

Figure 6.5 displays a practical example of a low-resolution photo directly retrieved from the Internet. While GPEN successfully restores the image, PULSE struggles with basic features. Due to lack of ground truth, no qualitative metrics are shown.

Figure 6.5 – In the wild BFR.

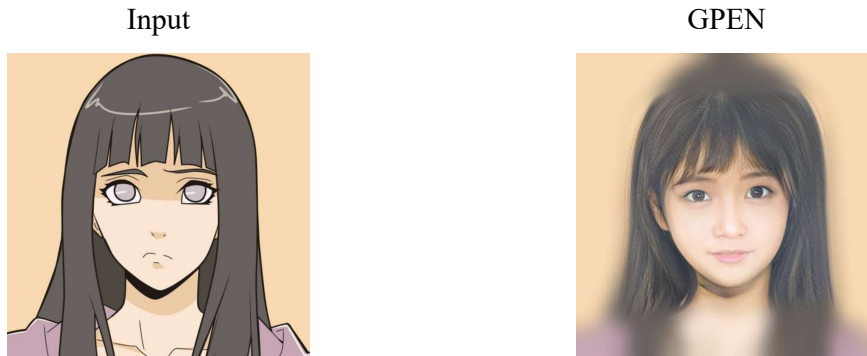


Source: the Author.

In order to operate as a blind method that simultaneously deblurs, upsamples, and denoises face images, PULSE requires that its inputs be made very small (e.g.,  $32 \times 32$ ), leading to loss of relevant details. A way to circumvent that is to make PULSE non-blind, i.e., use a specific, known degradation operation  $Y$  and minimize  $\|Y(G(z)) - I\|$ , where  $Y(x)$  computes  $(x \otimes k) \downarrow_s + n$ , instead of using the downscaling function  $D$ . However, even with small degradations, it may not be possible to recover identity without major changes in the original implementation, since actual inversion techniques would then be required (ABDAL et al. 2019). Also, as pointed out by TOV et al. (2021), optimization-based inversion converges to arbitrary points in latent space whereas outputs of CNN-based inversion lie in a tight space located within the natural image manifold (i.e., more realistic). Thus, this section deals only with the original blind implementation of PULSE, allowing for a fair comparison with GPEN.

Interestingly, GPEN can be used in image-to-image translation e.g., an anime face can be generated with the corresponding realistic cosplay like face as shown in Figure 6.6. To achieve that, simply simulate a degradation on the input face image (i.e., Gaussian blur) in such a way that the domain of the face image is no longer recognizable. GPEN will then be tricked into mapping that LR anime face image into any realistic face that shares the same set of colors, pose, and hairstyle.

Figure 6.6 – Image-to-image translation using GPEN.



Source: the Author.

In brief, GPEN and PULSE adopt very distinct GAN-based approaches. On the one hand, PULSE requires zero training time since it employs an off-the-shelf generative model, however during evaluation, it is much slower and the face identity is not preserved. On the other hand, GPEN requires assembling and fine-tuning a U-Net, nevertheless it is faster and often preserves identity. Finally, all these differences are further outlined by visual results.

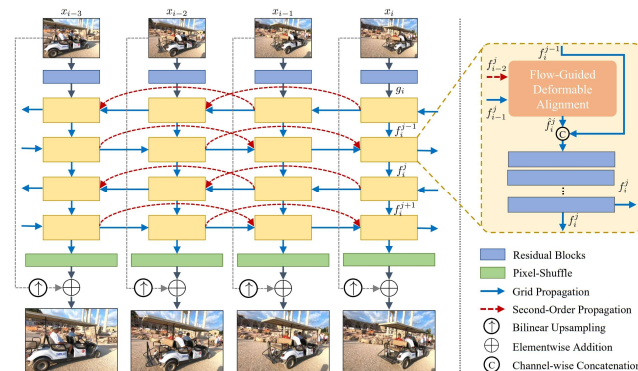
## 7 VIDEO RESTORATION

Unlike single-image restoration, video restoration can combine information from different frames, i.e., supporting frames, to better reconstruct any given frame, i.e., reference frame. Recently, VRT (LIANG et al., 2022) has obtained a new state-of-the-art in video-restoration which was previously held by BasicVSR++ (CHAN et al., 2022). As such, this chapter focuses on these two methods: BasicVSR++ is a recurrent model which uses previously reconstructed frames for subsequent frame reconstruction; and VRT is a transformer which processes its input frames in parallel, generating all outputs simultaneously.

### 7.1 BasicVSR++

BasicVSR++ is a higher order RNN (HORNN) (SOLTANI; JIANG, 2016) which descends from BasicVSR (CHAN et al., 2021a): an RNN which adopts typical 1D bidirectional propagation along with feature alignment. Unlike its ancestral, BasicVSR++ can better restore finer details and is more robust to occluded regions both of which can be attributed to arranging propagation as a 2D grid structure with higher-order connections as shown in Figure 7.1 and incorporating deformable alignment. While expressiveness is enhanced through repeated feature refinement within the grid structure, training is facilitated by the higher order (or second-order) propagation which improves gradient flow. Meanwhile, information is aligned implicitly through deformable convolution (CHAN et al., 2021b; DAI et al., 2017) which can better model geometric transformations. SPyNet (RANJAN; BLACK, 2017) is employed to estimate optical flow which is the apparent motion of scene points from the sequence of frames. This optical flow is used to guide deformable alignment with the goal of easing training.

Figure 7.1 – BasicVSR++ framework.



Source: Chan et al., (2022).

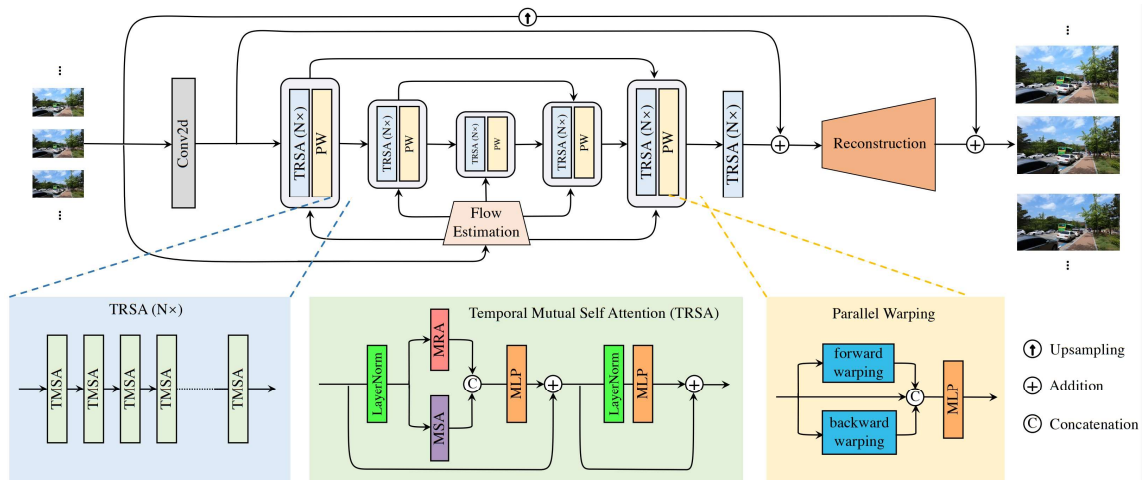
## 7.2 VRT

Given a batch of input frames, VRT (Video Restoration Transformer) begins by applying a single convolution over the entire batch to extract shallow features. Then, as shown in Figure 7.2, VRT follows a U-Net design to extract deep features where it employs two types of attention: self-attention for feature extraction and mutual attention for frame alignment. Whereas self-attention considers a single input signal  $X$  when computing  $Q$ ,  $K$ , and  $V$ , mutual-attention relates two different input signals  $X$  (for queries) and  $Y$  (for keys and values). Then, self/mutual-attention is computed as usual, i.e.,  $Q$  queries  $K$  to generate the attention map

$$M = \text{SoftMax}(QK^T / \sqrt{D}) \quad (7.1)$$

which is then used for a weighted sum of  $V$ , i.e.,  $\text{attention} = MV$ . In order to employ mutual attention effectively, VRT creates partitions of two frames each, i.e., a short clip, which are processed by TMSA as shown in Figure 7.2. Finally, a reconstruction model which depends on the image restoration task, e.g., deblurring or super-resolution, combines shallow and deep features to produce the output batch.

Figure 7.2 – VRT framework.



Source: Liang et al., (2022).

## 7.3 VRT vs. BasicVSR++

The quantitative comparison presented by Liang et al. (2022) shows that VRT scores higher PSNR/SSIM than BasicVSR++. Therefore, experiments below show visual results for



VRT only. However, Liang et al. (2022) also acknowledges that the 32-frame BasicVSR++ model is slightly faster than the VRT model trained on 16-frames.

Figure 7.3 shows six input frames from Vid4/BIX4/foilage dataset (LIU; SUN, 2014) which contains bicubic degradation using scale factor  $sf = 4$ . In addition, VRT model 001\_VRT\_videosr\_bi\_REDS\_6frames is used. Since VRT processes each frame in parallel which is very demanding in terms of computer memory, only some frames are used, i.e.,  $\{0, 7, 14, 21, 28, 35\}$ .

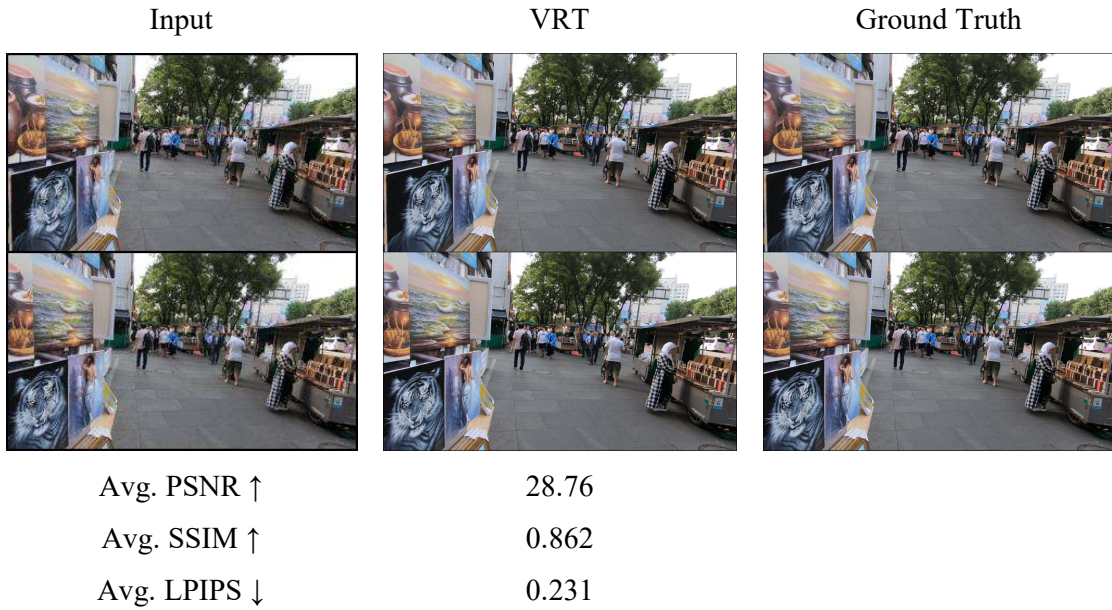
Figure 7.3 – Experiments on Vid4.



Source: the Author.

Figure 7.4 shows input frames 75 and 76 from REDS4/sharp\_bicubic/020 dataset (WANG et al., 2019).

Figure 7.4 – Experiments on REDS4.



Source: the Author.

As mentioned, there is a tradeoff between using these models: BasicVSR++ for speed and VRT for visual quality. In addition, VRT is memory intensive due to the parallel computation of feature maps. Despite that, such limitation could be overcome by redesigning VRT as a distributed application as discussed by Liang et al. (2022).

## 8 CONCLUSION

To make the best out of a neural network requires understanding both the individual parts (e.g., a  $3 \times 3$  convolutional layer) that make it as well as the interactions between these parts. Acquiring such understanding is facilitated by studying multiple different methods. However, another critical component is experimentation which entails designing a network, choosing appropriate datasets, and trying different training strategies. In spite of that, there are instances where not even the authors themselves fully understand their newly created architectures (e.g., the AdaIN operation in StyleGAN1 which was later replaced in StyleGAN2 due to characteristic artifacts in the generated images). In such cases, one may begin to speculate about the mechanics of the network.

While it can be very challenging to acquire a profound understanding of neural networks, several conclusions can be drawn when contemplating deep image restoration methods. Above all, there are design tendencies: (1) to aid training, models only learn the difference between (LQ, HQ) image pairs. Such difference is then added as a residual to the final output; (2) Although transformers require a large training corpus, incorporating them into a hybrid architecture often leads to the best results; and (3) domain-based image restoration is best achieved through generative models. In addition, existing limitations may hint at what is to come: (4) successfully deblurring arbitrary images requires taking both global and local dependencies into account. The results of DeepRFT and Uformer show there is still a large room for improvement; (5) SR methods usually operate on a small set of scale factors which does not capture real world degradations; (6) While deep non-blind methods have the potential to deal with a complete degradation model, the best way to do so remains disputable; (7) Priors help face restoration models, however recovering identity of highly degraded photos is still challenging; (8) Due the computational cost which rises with the height and width of frames, state-of-the-art video restoration methods are not ready for casual use yet.

As evidenced by SRRUNet, one can successfully fine-tune a denoising model into a super-resolution model. Such experiment has only taken place because a broader picture on the field of image restoration had already been previously established. In fact, taking a broader perspective on artificial intelligence problems often leads to successful and inspiring methods. For instance, without the original work on Generative Adversarial Networks which integrated the distinct realms of game theory and deep learning, there would be no StyleGANs. As discussed in Chapter 6, StyleGANs are the key component of face restoration methods. As a second example, consider Diffusion Models which have record-breaking performance in



generative modelling and are currently considered SOTA. Diffusion Models employ Nonequilibrium Thermodynamics which, at a first glance, seem to have little correlation with Deep Learning. A third example is the Vision Transformer which has been used in numerous works some of which have been discussed in this thesis. The Vision Transformer has emerged from the Transformer, an architecture originally meant for processing text. Whether one sees it or not, it is all connected. A broader view is what enables seeing these connections.

## 8.1 Research Directions

Upon studying multiple methods, several ideas which could lead to improvements come to mind. To start with, it worth questioning the design of Res FFT-Conv Blocks which employ  $1 \times 1$  convolutions in the FFT stream. A  $1 \times 1$  convolution is the same as multiplying each channel of the feature map by a single learnable scalar. However, if the aim is to better leverage global context, then perhaps it would be more useful to employ an MLP instead, since MLPs apply a different scalar for each pixel. In this case, because MLPs require the size of the image to be previously specified, the only requirement would be resizing the feature maps before forwarding them to the MLP. In addition, the U-Net architecture of Uformer could be enhanced with more residual connections, thus mimicking DRUNet while also keeping the LeWin blocks. Moreover, since both SROOE and HAT successfully deal with scale factor 4, it would be interesting to train these models using larger scale factors (e.g., 16). Meanwhile, DRUNet which is non-blind, could be made blind by employing a second network which estimates the noise level for each pixel. As for PULSE and GPEN, in order to stand a chance at recovering identity of highly degraded photos, there could be an option to provide additional input photos of the same person in which case a face recognition network could be used to compute identity loss. Regarding VRT, its memory demands could be reduced by reducing the overall number of channels for each feature map while increasing the depth of the network in order to maintain efficacy.

Even though it is interesting that gradient descent can be used for image deconvolution, there are much faster methods. As discussed in Chapter 5, USRNet uses an equation which employs frequency domain properties. However, since USRNet is iterative, it is relatively slow when compared to other neural networks. Furthermore, USRNet deals with multiple degradations all at once and, as such, the network is rather generic. Therefore, with the aim of conceiving a faster network which would also yield higher quality results, different modules

could be trained to work in conjunction with the equation employed by USRNet: a module which removes noise (e.g., DRUNet), a module which removes deconvolution artifacts, a module which increases resolution, and, optionally, a module which generates hyperparameters. Thus, the iterative approach would no longer be required and each module would also be more specific and, therefore, expected to lead to better results. Of course, the complete network could also be further fine-tuned so that each module learns to adapt to each other.

In order to enable fair comparison with methods such as SROOE and HAT, SRRUNet could be trained using only a specific scale factor. In addition, instead of training with a single loss function, the OOE framework of SROOE could be used. Moreover, in order to facilitate reproducibility which is a major principle of the scientific method, training could start using randomly initialized weights using a specific seed for the pseudo-random number generator instead of fine-tuning DRUNet which requires downloading the original pretrained model. Finally, it would be interesting to also change the network architecture, perhaps switching from a Residual U-Net to a Residual Uformer as aforementioned.

## REFERENCES

- ABDAL, R.; QIN, Y.; WONKA, P. Image2stylegan: How to embed images into the stylegan latent space? In: **2019 IEEE/CVF International Conference on Computer Vision (ICCV)**. [S.l.: s.n.], 2019. p. 4431–4440.
- ABUOLAIM, A.; BROWN, M. S. **Defocus Deblurring Using Dual-Pixel Data**. arXiv, 2020. Available from Internet: <<https://arxiv.org/abs/2005.00305>>.
- AGUSTSSON, E.; TIMOFTE, R. Ntire 2017 challenge on single image super-resolution: Dataset and study. In: **2017 IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)**. [S.l.: s.n.], 2017. p. 1122–1131.
- ANWAR, S.; KHAN, S.; BARNES, N. A deep journey into super-resolution: A survey. **ACM Computing Surveys (ACMCSUR)**, Association for Computing Machinery (ACM), New York, NY, USA, v. 53, n. 3, may 2020. ISSN 0360-0300.
- ARORA, S.; COHEN, N.; HAZAN, E. **On the Optimization of Deep Networks: Implicit Acceleration by Overparameterization**. arXiv, 2018. Available from Internet: <<https://arxiv.org/abs/1802.06509>>.
- BEVILACQUA, M. et al. Low-complexity single-image super-resolution based on nonnegative neighbor embedding. In: **Proceedings of the British Machine Vision Conference**. [S.l.]: BMVA Press, 2012. p. 135.1–135.10. ISBN 1-901725-46-4.
- CAO, J. et al. Do-conv: Depthwise over-parameterized convolutional layer. **IEEE Transactions on Image Processing**, v. 31, p. 3726–3736, 2022.
- CHAN, K. C. et al. Basicvsr: The search for essential components in video superresolution and beyond. In: **Proceedings of the IEEE conference on computer vision and pattern recognition**. [S.l.: s.n.], 2021.
- CHAN, K. C. et al. Understanding deformable alignment in video super-resolution. In: **AAAI Conference on Artificial Intelligence**. [S.l.: s.n.], 2021.
- CHAN, K. C. et al. BasicVSR++: Improving video super-resolution with enhanced propagation and alignment. In: **IEEE Conference on Computer Vision and Pattern Recognition**. [S.l.: s.n.], 2022.
- CHEN, X. et al. Activating more pixels in image super-resolution transformer. **arXiv preprint arXiv:2205.04437**, 2022.
- CHEN, Y.; POCK, T. Trainable nonlinear reaction diffusion: A flexible framework for fast and effective image restoration. **IEEE Transactions on Pattern Analysis and Machine Intelligence**, v. 39, n. 6, p. 1256–1272, 2017.
- CHO, S.-J. et al. **Rethinking Coarse-to-Fine Approach in Single Image Deblurring**. arXiv, 2021. Available from Internet: <<https://arxiv.org/abs/2108.05054>>.

CHOI, Y. et al. Stargan v2: Diverse image synthesis for multiple domains. In: **Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition**. [S.l.: s.n.], 2020.

DAI, J. et al. Deformable convolutional networks. In: **2017 IEEE International Conference on Computer Vision (ICCV)**. [S.l.: s.n.], 2017. p. 764–773.

DENG, J. et al. Imagenet: A large-scale hierarchical image database. In: **2009 IEEE Conference on Computer Vision and Pattern Recognition**. [S.l.: s.n.], 2009. p. 248–255.

DEVLIN, J. et al. **BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding**. arXiv, 2018. Available from Internet: <<https://arxiv.org/abs/1810.04805>>.

DONG, C. et al. Learning a deep convolutional network for image super-resolution. In: FLEET, D. et al. (Ed.). **Computer Vision – ECCV 2014**. Cham: Springer International Publishing, 2014. p. 184–199. ISBN 978-3-319-10593-2.

DONG, C.; LOY, C. C.; TANG, X. **Accelerating the Super-Resolution Convolutional Neural Network**. arXiv, 2016. Available from Internet: <<https://arxiv.org/abs/1608.00367>>.

DOSOVITSKIY, A. et al. **An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale**. arXiv, 2020. Available from Internet: <<https://arxiv.org/abs/2010.11929>>.

FORTUNATO, H. E.; OLIVEIRA, M. M. Fast high-quality non-blind deconvolution using sparse adaptive priors. **The Visual Computer**, v. 30, n. 6-8, p. 661–671, 2014. ISSN 0178-2789.

GASTAL, E. S. L.; OLIVEIRA, M. M. Domain transform for edge-aware image and video processing. **ACM Trans. Graph.**, Association for Computing Machinery, New York, NY, USA, v. 30, n. 4, jul 2011. ISSN 0730-0301. Available from Internet: <<https://doi.org/10.1145/2010324.1964964>>.

GONG, D. et al. **Learning Deep Gradient Descent Optimization for Image Deconvolution**. arXiv, 2018. Available from Internet: <<https://arxiv.org/abs/1804.03368>>.

GONZALEZ, R. C.; WOODS, R. E. **Digital Image Processing (3rd Edition)**. USA: Prentice-Hall, Inc., 2006. ISBN 013168728X.

GOODFELLOW, I. J. et al. **Generative Adversarial Networks**. arXiv, 2014. Available from Internet: <<https://arxiv.org/abs/1406.2661>>.

HE, K. et al. Deep residual learning for image recognition. In: **2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)**. [S.l.: s.n.], 2016. p. 770–778.

HORNIK, K. Approximation capabilities of multilayer feedforward networks. **Neural Networks**, v. 4, n. 2, p. 251–257, 1991. ISSN 0893-6080. Available from Internet: <<https://www.sciencedirect.com/science/article/pii/089360809190009T>>.

HUANG, J.-B.; SINGH, A.; AHUJA, N. Single image super-resolution from transformed self-exemplars. In: **2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)**. [S.l.: s.n.], 2015. p. 5197–5206.

KARRAS, T. et al. **Progressive Growing of GANs for Improved Quality, Stability, and Variation**. arXiv, 2017. Available from Internet: <<https://arxiv.org/abs/1710.10196>>.

KARRAS, T.; LAINE, S.; AILA, T. **A Style-Based Generator Architecture for Generative Adversarial Networks**. arXiv, 2018. Available from Internet: <<https://arxiv.org/abs/1812.04948>>.

KARRAS, T. et al. Analyzing and improving the image quality of StyleGAN. In: **Proc. CVPR**. [S.l.: s.n.], 2020.

KINGMA, D. P.; BA, J. **Adam: A Method for Stochastic Optimization**. arXiv, 2014. Available from Internet: <<https://arxiv.org/abs/1412.6980>>.

LIANG, J. et al. Vrt: A video restoration transformer. **arXiv preprint arXiv:2201.12288**, 2022.

LIANG, J. et al. Swinir: Image restoration using swin transformer. **arXiv preprint arXiv:2108.10257**, 2021.

LIM, B. et al. Enhanced deep residual networks for single image super-resolution. In: **The IEEE Conference on Computer Vision and Pattern Recognition (CVPR) Workshops**. [S.l.: s.n.], 2017.

LIU, C.; SUN, D. On bayesian adaptive video super resolution. **IEEE Transactions on Pattern Analysis and Machine Intelligence**, v. 36, n. 2, p. 346–360, 2014.

LIU, Z. et al. Swin transformer: Hierarchical vision transformer using shifted windows. In: **Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)**. [S.l.: s.n.], 2021.

LUO, X. et al. Time-travel rephotography. **ACM Trans. Graph.**, Association for Computing Machinery, New York, NY, USA, v. 40, n. 6, dec 2021. ISSN 0730-0301. Available from Internet: <<https://doi.org/10.1145/3478513.3480485>>.

MA, K. et al. Waterloo exploration database: New challenges for image quality assessment models. **IEEE Transactions on Image Processing**, v. 26, n. 2, p. 1004–1016, 2017.

MAO, X. et al. Deep residual fourier transformation for single image deblurring. In: **arXiv:2111.11745**. [S.l.: s.n.], 2021.

MARTIN, D. et al. A database of human segmented natural images and its application to evaluating segmentation algorithms and measuring ecological statistics. In: **Proceedings Eighth IEEE International Conference on Computer Vision. ICCV 2001**. [S.l.: s.n.], 2001. v. 2, p. 416–423 vol.2.

MATSUI, Y. et al. Sketch-based manga retrieval using manga109 dataset. **Multimedia Tools Appl.**, Kluwer Academic Publishers, USA, v. 76, n. 20, p. 21811–21838, oct 2017. ISSN 1380-7501. Available from Internet: <<https://doi.org/10.1007/s11042-016-4020-z>>.

MENON, S. et al. **PULSE: Self-Supervised Photo Upsampling via Latent Space Exploration of Generative Models**. arXiv, 2020. Available from Internet: <<https://arxiv.org/abs/2003.03808>>.

NAH, S. et al. Ntire 2019 challenge on video deblurring and super-resolution: Dataset and study. In: **2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)**. [S.l.: s.n.], 2019. p. 1996–2005.

NAH, S.; KIM, T. H.; LEE, K. M. Deep multi-scale convolutional neural network for dynamic scene deblurring. In: **The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)**. [S.l.: s.n.], 2017.

PAN, J. et al. Kernel estimation from salient structure for robust motion deblurring. **Signal Processing: Image Communication**, v. 28, n. 9, p. 1156–1170, 2013. ISSN 0923-5965. Available from Internet: <<https://www.sciencedirect.com/science/article/pii/S0923596513000623>>.

PAN, J. et al. Blind image deblurring using dark channel prior. In: **2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)**. [S.l.: s.n.], 2016. p. 1628–1636.

PARK, S. H.; MOON, Y. S.; CHO, N. I. **Perception-Oriented Single Image SuperResolution using Optimal Objective Estimation**. arXiv, 2022. Available from Internet: <<https://arxiv.org/abs/2211.13676>>.

PATASHNIK, O. et al. Styleclip: Text-driven manipulation of stylegan imagery. In: **Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)**. [S.l.: s.n.], 2021. p. 2085–2094.

POULI, T.; REINHARD, E.; CUNNINGHAM, D. W. **Image Statistics in Visual Computing**. 1st. ed. USA: A. K. Peters, Ltd., 2013. ISBN 1568817258.

RADFORD, A. et al. **Learning Transferable Visual Models From Natural Language Supervision**. arXiv, 2021. Available from Internet: <<https://arxiv.org/abs/2103.00020>>.

RANJAN, A.; BLACK, M. J. Optical flow estimation using a spatial pyramid network. In: **2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)**. [S.l.: s.n.], 2017. p. 2720–2729.

RIM, J. et al. Real-world blur dataset for learning and benchmarking deblurring algorithms. In: **Proceedings of the European Conference on Computer Vision (ECCV)**. [S.l.: s.n.], 2020.

ROGOZHNIKOV, A. Einops: Clear and reliable tensor manipulations with einstein-like notation. In: **International Conference on Learning Representations**. [s.n.], 2022. Available from Internet: <<https://openreview.net/forum?id=oapKSVM2bcj>>.

RONNEBERGER, O.; FISCHER, P.; BROX, T. **U-Net: Convolutional Networks for Biomedical Image Segmentation**. arXiv, 2015. Available from Internet: <<https://arxiv.org/abs/1505.04597>>.

SHAN, Q.; JIA, J.; AGARWALA, A. High-quality motion deblurring from a single image. **ACM Trans. Graph.**, Association for Computing Machinery, New York, NY, USA, v. 27, n. 3, p. 1–10, aug 2008. ISSN 0730-0301. Available from Internet: <<https://doi.org/10.1145/1360612.1360672>>.

SHEN, Z. et al. Human-aware motion deblurring. In: **IEEE International Conference on Computer Vision**. [S.l.: s.n.], 2019.

SHI, W. et al. Real-time single image and video super-resolution using an efficient subpixel convolutional neural network. **CoRR**, abs/1609.05158, 2016. Available from Internet: <<http://arxiv.org/abs/1609.05158>>.

SIMONYAN, K.; ZISSERMAN, A. Very deep convolutional networks for large-scale image recognition. In: . [S.l.]: Computational and Biological Learning Society, 2015. p. 1–14.

SOLTANI, R.; JIANG, H. **Higher Order Recurrent Neural Networks**. arXiv, 2016. Available from Internet: <<https://arxiv.org/abs/1605.00064>>.

TIMOFTE, R. et al. Ntire 2017 challenge on single image super-resolution: Methods and results. In: **2017 IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)**. [S.l.: s.n.], 2017. p. 1110–1121.

TOV, O. et al. Designing an encoder for stylegan image manipulation. **ACM Trans. Graph.**, Association for Computing Machinery, New York, NY, USA, v. 40, n. 4, jul 2021. ISSN 0730-0301. Available from Internet: <<https://doi.org/10.1145/3450626.3459838>>.

ULYANOV, D.; VEDALDI, A.; LEMPITSKY, V. **Instance Normalization: The Missing Ingredient for Fast Stylization**. arXiv, 2016. Available from Internet: <<https://arxiv.org/abs/1607.08022>>.

ULYANOV, D.; VEDALDI, A.; LEMPITSKY, V. Deep image prior. **arXiv:1711.10925**, 2017.

VASWANI, A. et al. **Attention Is All You Need**. arXiv, 2017. Available from Internet: <<https://arxiv.org/abs/1706.03762>>.

WANG, X. et al. Edvr: Video restoration with enhanced deformable convolutional networks. In: **The IEEE Conference on Computer Vision and Pattern Recognition (CVPR) Workshops**. [S.l.: s.n.], 2019.

WANG, X. et al. **Recovering Realistic Texture in Image Super-resolution by Deep Spatial Feature Transform**. arXiv, 2018. Available from Internet: <<https://arxiv.org/abs/1804.02815>>.

WANG, X. et al. Esrgan: Enhanced super-resolution generative adversarial networks. In: **The European Conference on Computer Vision Workshops (ECCVW)**. [S.l.: s.n.], 2018.

WANG, Y. et al. Flickr1024: A large-scale dataset for stereo image super-resolution. In: **International Conference on Computer Vision Workshops**. [S.l.: s.n.], 2019. p. 3852–3857.

WANG, Z. et al. Image quality assessment: from error visibility to structural similarity. **IEEE Transactions on Image Processing**, v. 13, n. 4, p. 600–612, 2004.

WANG, Z. et al. Uformer: A general u-shaped transformer for image restoration. In: **Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)**. [S.l.: s.n.], 2022. p. 17683–17693.

WIENER, N. Index. In: \_\_\_\_\_. **Extrapolation, Interpolation, and Smoothing of Stationary Time Series: With Engineering Applications**. [S.l.: s.n.], 1964. p. 161–163.

XUE, T. et al. Video enhancement with task-oriented flow. **International Journal of Computer Vision**, Springer Science and Business Media LLC, v. 127, n. 8, p. 1106–1125, feb 2019. Available from Internet: <<https://doi.org/10.1007%2Fs11263-018-01144-2>>.

YANG, T. et al. Gan prior embedded network for blind face restoration in the wild. In: **IEEE Conference on Computer Vision and Pattern Recognition (CVPR)**. [S.l.: s.n.], 2021.

YANG, X. An overview of the attention mechanisms in computer vision. **Journal of Physics: Conference Series**, IOP Publishing, v. 1693, n. 1, p. 012173, dec 2020. Available from Internet: <<https://dx.doi.org/10.1088/1742-6596/1693/1/012173>>.

YASARLA, R.; PERAZZI, F.; PATEL, V. M. Deblurring face images using uncertainty guided multi-stream semantic networks. **IEEE Transactions on Image Processing**, v. 29, p. 6251–6263, 2020.

YI, P. et al. Progressive fusion video super-resolution network via exploiting non-local spatio-temporal correlations. In: **2019 IEEE/CVF International Conference on Computer Vision (ICCV)**. [S.l.: s.n.], 2019. p. 3106–3115.

ZEYDE, R.; ELAD, M.; PROTTER, M. On single image scale-up using sparse representations. In: BOISSONNAT, J.-D. et al. (Ed.). **Curves and Surfaces**. Berlin, Heidelberg: Springer Berlin Heidelberg, 2012. p. 711–730. ISBN 978-3-642-27413-8.

ZHANG, K.; GOOL, L. V.; TIMOFTE, R. Deep unfolding network for image superresolution. In: **IEEE Conference on Computer Vision and Pattern Recognition**. [S.l.: s.n.], 2020. p. 3217–3226.

ZHANG, K. et al. Plug-and-play image restoration with deep denoiser prior. **IEEE Transactions on Pattern Analysis and Machine Intelligence**, v. 44, n. 10, p. 6360–6376, 2021.

ZHANG, K. et al. **Deep Image Deblurring: A Survey**. arXiv, 2022. Available from Internet: <<https://arxiv.org/abs/2201.10700>>.



ZHANG, R. et al. The unreasonable effectiveness of deep features as a perceptual metric. In: **CVPR**. [S.l.: s.n.], 2018.

ZHAO, H. et al. Loss functions for image restoration with neural networks. **IEEE Transactions on Computational Imaging**, v. 3, n. 1, p. 47–57, 2017.

ZHAO, N. et al. Fast single image super-resolution using a new analytical solution for  $\ell_2 - \ell_2$  problems. **IEEE Transactions on Image Processing**, v. 25, n. 8, p. 3683–3697, 2016.